

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号  
特開2000-75785  
(P2000-75785A)

(43)公開日 平成12年3月14日(2000.3.14)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード(参考)
G 0 9 C 1/00	6 1 0	G 0 9 C 1/00	6 1 0 B
H 0 4 L 9/06		H 0 4 L 9/00	6 1 1 A

審査請求 未請求 請求項の数15 O L (全 20 頁)

(21)出願番号	特願平10-240844	(71)出願人	000005223 富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番1号
(22)出願日	平成10年8月26日(1998.8.26)	(72)発明者	長谷部 高行 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72)発明者	岡田 壮一 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(74)代理人	100074099 弁理士 大菅 義之 (外1名)

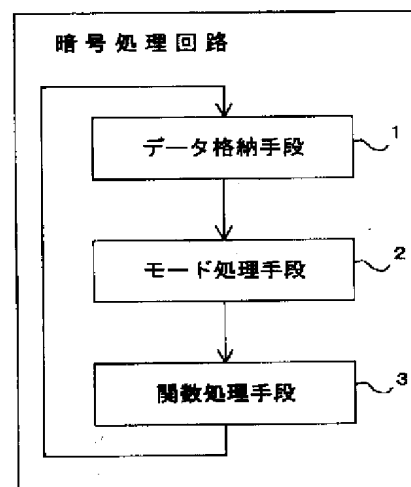
(54)【発明の名称】 高速暗号処理回路および処理方法

(57)【要約】

【課題】 DES暗号の操作モードとしてのCBCモードとCFBモードとの両方を実行できる暗号処理回路を提供し、かつその回路における処理遅延をできるだけ削減する。

【解決手段】 ブロック暗号化処理に用いられる初期ベクトル、処理中間値、または処理最終値を格納する手段1と、ブロック暗号化処理内で用いられる関数処理を実行する手段2と、手段1と手段2との間に備えられ、ブロック暗号化の操作モードに対応する処理を実行する手段3とを備える。

第1の実施形態に対応する  
原理構成ブロック図



## 【特許請求の範囲】

【請求項1】 ブロック暗号化の処理を行う暗号処理回路において、

ブロック暗号化処理に用いられる初期ベクトル、処理中間値、または処理最終結果を格納するデータ格納手段と、

ブロック暗号化処理内で用いられる関数処理を実行する関数処理手段と、

該データ格納手段と関数処理手段との間に備えられ、ブロック暗号化の操作モードに対応する処理を実行するモード処理手段とを備えることを特徴とする高速暗号処理回路。

【請求項2】 前記モード処理手段が排他的論理和回路によって構成されることを特徴とする請求項1記載の高速暗号処理回路。

【請求項3】 前記ブロック暗号がDES暗号であり、前記操作モードがCFBモードであることを特徴とする請求項1、または2記載の高速暗号処理回路。

【請求項4】 前記関数処理手段が、前記DES暗号へのブロックデータの暗号化における前記関数処理としての非線形関数を用いる16段の変換を複数回に分けて実行することを特徴とする請求項3記載の高速暗号処理回路。

【請求項5】 前記関数処理手段が、前記複数回の最後の1回を除く毎回の実行結果を前記データ格納手段に出力することを特徴とする請求項4記載の高速暗号処理回路。

【請求項6】 前記関数処理手段による、前記複数回の最終回の処理結果を用いて、入力ブロックデータに対する暗号化ブロックデータを出力する暗号化データ出力手段を更に備えることを特徴とする請求項4記載の高速暗号処理回路。

【請求項7】 前記関数処理手段による、前記複数回の最初の回の実行の前に、前記モード処理手段がモード処理を実行し、該実行結果を該関数処理手段に出力すると共に、

該モード処理の実行結果の一部を格納するモード処理結果格納手段を更に備えることを特徴とする請求項4記載の高速暗号処理回路。

【請求項8】 前記関数処理手段による、前記複数回の最後の回の実行時に、前記モード処理結果格納手段が保持しているデータを前記データ格納手段に出力すると共に、該関数処理手段が該実行結果の一部を該データ格納手段に出力することを特徴とする請求項7記載の高速暗号処理回路。

【請求項9】 前記ブロック暗号がDES暗号であり、前記操作モードがCBCモードであることを特徴とする請求項1、または2記載の高速暗号処理回路。

【請求項10】 前記関数処理手段が、前記DES暗号へのブロックデータの暗号化における前記関数処理とし

ての非線形関数を用いる16段の変換を複数回に分けて実行することを特徴とする請求項9記載の高速暗号処理回路。

【請求項11】 前記関数処理手段が、前記複数回の毎回の実行結果を前記データ格納手段に出力することを特徴とする請求項9記載の高速暗号処理回路。

【請求項12】 前記関数処理手段による前記複数回の最初の回の実行の前に、前記モード処理手段が入力されるブロックデータと前記格納手段に格納されているデータとを用いてモード処理を実行し、該モード処理の結果を該関数処理手段に出力することを特徴とする請求項9記載の高速暗号処理回路。

【請求項13】 ブロック暗号化の処理を行う暗号処理回路において、

ブロック暗号化処理に用いられる初期ベクトル、処理中間値、または処理最終結果を格納するデータ格納手段と、

ブロック暗号化処理内で用いられる関数処理を実行する関数処理手段と、

該データ格納手段と関数処理との間に備えられ、ブロック暗号化の操作モードに対応する第1の処理を実行する第1のモード処理手段と、

該関数処理手段とデータ格納手段との間に備えられ、ブロック暗号化の操作モードに対応する第2の処理を実行する第2のモード処理手段とを備えることを特徴とする高速暗号処理回路。

【請求項14】 ブロック暗号化の処理を行う暗号処理方法において、

1つのブロックデータの暗号化処理の終了時点で処理結果を出力すると共に、初期ベクトルを格納していたレジスタに該暗号化処理における処理中間値、または処理最終結果を格納し、

該1つのブロックデータ、または該1つのブロックに続く次のブロックのデータとレジスタに格納されたデータとの間で該ブロック暗号化の操作モードに対応する演算を実行し、

該演算結果を、該1つのブロックに続く次のブロックデータに対する暗号化処理における前記初期ベクトルに相当する値として用いて、該次のブロックデータに対する暗号化処理を行うことを特徴とする暗号処理方法。

【請求項15】 前記ブロック暗号がDES暗号であり、前記操作モードがCFBモード、またはCBCモードであることを特徴とする請求項14記載の暗号処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はデジタルデータの暗号化方式に関し、更に詳しくは代表的なブロック暗号としてのDES暗号のCBCモードとCFBモードの処理を実行する高速暗号処理回路に関する。

10

20

30

40

50

## 【0002】

【従来の技術】近年デジタルデータの暗号化処理が注目されており、暗号化処理の対象となるデータも増加している。例えば動画データデータの暗号化処理や、サーバにおけるトランザクション処理の要求が出てきており、暗号化処理自体にも高速化が求められている。

【0003】デジタルデータの暗号化方式としては、例えば64ビットを1つのブロックとしてブロック毎に暗号化を行うブロック暗号方式と、例えば1ビットずつ逐次的に暗号化を行うストリーム暗号方式とがある。

【0004】本発明が対象とする暗号化方式はブロック暗号であり、その中でも最も代表的なDES暗号である。DES（データ エンクリプション スタンダード）は1977年1月にアメリカ商務省標準局によって定められたデータ暗号標準であり、現在最も広範に用いられている暗号アルゴリズムである。

【0005】DES暗号に対しては、操作モードとして4つのモードが定められている。そのモードはECB（エレクトロニック コード ブロック）、CBC（サイファブロック チェイニング）、CFB（サイファフィードバック）およびOFB（アウトプット フィードバック）の4つである。

【0006】これらの4つのモードのうち、本発明に直接関連があるのはCBC、およびCFBの2つのモードであるため、これらの2つの操作モードにおける暗号化および復号の処理について、図24～図27を用いて説明する。

【0007】図24はCBCモードにおける暗号化および復号の基本説明図である。このモードでは64ビットに分割された平文系列 $M_1$ 、 $M_2$ 、 $M_3$ 、・・・が入力されるが、まず最初の64ビットとしての $M_1$ に対しては初期ベクトルIVとの排他的論理和51がとられ、その結果に対して暗号化処理E52が実行されて暗号文系列 $C_1$ 、 $C_2$ 、 $C_3$ 、・・・のうちの最初の64ビットとしての $C_1$ が得られる。ここで暗号化処理としてのE52は初期転置IP、f関数を用いた同一構造の16段の変換、左右32ビットの入替え、および逆転置IP<sup>-1</sup>を含むものである。

【0008】次の64ビットとしての $M_2$ の入力に対しては、暗号文の最初の64ビットとしての $C_1$ との排他的論理和56がとられ、その結果に対して暗号化処理E57が行われ、暗号文の次の64ビットとしての $C_2$ が得られる。以下同様にして暗号文が64ビットずつ作成される。

【0009】復号処理においては、暗号文の最初の64ビットとしての $C_1$ の入力に対して復号処理D53が実行される。この処理は暗号化処理E52と同様に初期転置IP、f関数を用いた16段の変換、左右32ビットの入替え、および逆転置IP<sup>-1</sup>を含むものであるが、暗号化処理と異なって16段の変換においてキー系列が逆

の順序で用いられる。復号処理D53の結果に対して、初期ベクトルIVとの排他的論理和54がとられ、平文の最初の64ビットとしての $M_1$ が得られる。

【0010】暗号文の次の64ビットとしての $C_2$ の入力に対して復号処理D58が行われた後に、その結果に対して暗号文の最初の64ビットとしての $C_1$ との排他的論理和59がとられ、平文の次の64ビットとしての $M_2$ が得られる。以下同様にして復号結果としての平文が作成される。

10 【0011】図25はCFBモードの動作の基本説明図である。CFBモードは乱数発生動作を行うものであり、平文は1以上64以下であるkビットずつに分割され、その系列は図24におけると同様に $M_1$ 、 $M_2$ 、 $M_3$ 、・・・とされる。

20 【0012】平文の、例えば最初の8ビットとしての $M_1$ の入力に対応して、レジスタ66に格納されている初期ベクトルIVの暗号化処理E67が行われる。この暗号化処理そのものは図24のCBCモードにおけると同じである。暗号化処理の結果の64ビットのうち左側の8ビットだけが取り出され、 $M_1$ との排他的論理和68がとられ、暗号文の最初の8ビット $C_1$ が出力されると共に、その暗号文としての8ビット $C_1$ はレジスタ66の最も右側に格納される。すなわちレジスタ66に格納されていた初期ベクトルIVは8ビット左シフトされて、左側の8ビットが捨てられ、最も右側に $C_1$ が格納されることになる。

30 【0013】平文の次の8ビット $M_2$ の入力に対応して、レジスタ66の格納内容に対する暗号化処理E67が行われ、その結果の64ビットのうちの左8ビットと $M_2$ との排他的論理和68がとられ、暗号文の次の8ビット $C_2$ として出力されると共に、その8ビットはレジスタ66の最も右側に格納される。すなわちレジスタ66の格納内容は8ビット左シフトされ、初期ベクトルIVのうち更に8ビットが捨てられて、レジスタ66の最も右側に $C_2$ が格納されることになる。暗号文系列の次の8ビット $C_3$ 以下については全く同様の動作が行われる。

40 【0014】CFBモードの復号処理においては、まず暗号文の最初の8ビット $C_1$ の入力に対応して、レジスタ70に格納されていた初期ベクトルIVの暗号化処理E71が行われる。図24のCBCモードと異なり、復号処理においても処理Dではなく、処理Eが用いられる。この処理の結果の64ビットのうち左8ビットが取り出され、暗号文の $C_1$ との排他的論理和72がとられ、平文の最初の8ビット $M_1$ が出力される。それと同時に暗号化処理におけると同様に、レジスタ70の内容は8ビット左シフトされ、最も右側に暗号文の8ビット $C_1$ が格納される。

50 【0015】暗号文系列の次の8ビット $C_2$ の入力に対応する処理は全く同様である。すなわちレジスタ70の

内容に対して暗号化処理E 7 1が行われ、その結果の64ビット中の左8ビットとC<sub>2</sub>との排他的論理和7 2がとられ、平文の次の8ビットM<sub>2</sub>が得られると共に、レジスタ7 0の内容は8ビット左シフトされ、暗号文の8ビットC<sub>2</sub>がレジスタ7 0の最も右側に格納される。以下同様にして、暗号文系列C<sub>3</sub>、・・・の復号処理が実行される。

【0016】図2 6はCBCモードの暗号処理回路の従来例である。同図(a)は暗号化回路を示す。同図において、図2 4におけると同様に平文の最初の8ビットM<sub>1</sub>が10 入力されると、レジスタ7 5に格納されている初期ベクトルI Vとの排他的論理和がE X O R回路7 6によってとられ、その結果はf関数7 7に入力される。

【0017】f関数7 7としては、f関数を用いた1 6段の変換を全て行うために1 6段の変換回路を備えてもよいが、例えばハードウェア量を減らすために、4段の変換回路を備え、処理の中間結果をレジスタ7 5およびE X O R回路7 6を介して4回ループさせることによって、1 6段の変換を実現することもできる。この場合、そのループの処理の間は、E X O R回路7 6に対する入20 力の値を64ビット全てに対して例えば“0”とすることによって、レジスタ7 5の格納内容をそのままf関数7 7に出力することができる。

【0018】なおここでは簡単のためにf関数7 7のみを示したが、図2 4の暗号化処理E 5 2に含まれる初期転置I Pは例えば第1段目のf関数の処理に含まれるものとし、また最後の左右3 2ビットの入替え、および逆転置I P<sup>-1</sup>は1 6段目のf関数の処理に含まれるもの25 と考えることができる。

【0019】そして4回目のループの終了時点でf関数7 7から出力され、レジスタ7 5に格納される64ビットは、暗号文の最初のブロックC<sub>1</sub>として出力されると共に、入力平文の次のブロックM<sub>2</sub>の入力時点でM<sub>2</sub>と排他的論理和をとるために用いられる。以下の処理は図2 4に対する説明と同様であるので、その説明を省略する。

【0020】図2 6 (b)はCBC復号回路の従来例である。同図において暗号文の最初のブロックC<sub>1</sub>が30 入力されると、その入力値はセクタ8 0を介してf関数8 1に与えられる。

【0021】このf関数8 1は図2 4における復号処理D 5 3を実行するためのものであり、(a)と同様に4段の変換回路を備えるものとする、レジスタ8 2、セクタ8 0を介する4回のループの動作の終了後に入力暗号文ブロックC<sub>1</sub>の復号結果をレジスタ8 2に出力することになり、また第1段目の変換には初期転置I P、1 6段目の変換には左右3 2ビットの変換と逆転置I P<sup>-1</sup>を含むものとする。

【0022】レジスタ8 2に格納された暗号文の最初のブロックC<sub>1</sub>の復号結果はE X O R回路8 4に入力さ40

れ、レジスタ8 3に格納されていた初期ベクトルI Vとの排他的論理和がとられ、平文の最初のブロックM<sub>1</sub>として出力される。この時次のブロックC<sub>2</sub>の復号結果と排他的論理和をとるために、入力暗号文の最初のブロックC<sub>1</sub>がレジスタ8 3に格納される。暗号文の次のブロックC<sub>2</sub>の入力に対する復号処理は図2 4におけると同様に実行されるので、以下その説明を省略する。

【0023】図2 7はCFBモードの暗号処理回路の従来例である。同図(a)は暗号化回路、(b)は復号回路を示す。図2 7 (a)において平文の最初の8ビットM<sub>1</sub>の入力に対応して、まずレジスタ8 6に格納されている初期ベクトルI Vの値がf関数8 7に出力される。

【0024】f関数8 7は図2 5における暗号化処理E 6 7、すなわち図2 4における暗号化処理E 5 2と同様の処理を行うものであり、図2 6におけると同様に例えば4段の変換回路が備えられる。セクタ9 0およびレジスタ8 6を介する4回のループの処理による1 6段の変換(図2 6におけると同様に初期転置、左右3 2ビットの入替え、および逆転置を含む)の処理結果としての64ビットのうち、左8ビットのみがE X O R回路9 1に出力され、入力平文の8ビットM<sub>1</sub>と排他的論理和がとれら、暗号文の最初のブロックC<sub>1</sub>として出力される。

【0025】前述のように、レジスタ8 6に格納されていた初期ベクトルI VはM<sub>1</sub>の入力に対応してf関数8 7に出力されるが、この時その内容は8ビット左にシフトされ、すなわち左の8ビットが捨てられて、レジスタ8 8に格納される。そのビット数は5 6ビットである。f関数8 7の4回目のループの後の出力のうち、最も左側の8ビットのみがE X O R回路8 9に入力され、入力M<sub>1</sub>との排他的論理和がとられ、その結果の8ビットが最も右側に、レジスタ8 8に格納されている5 6ビットが左側に配置される形式で、セクタ9 0を介してレジスタ8 6に64ビットのデータとして格納される。この64ビットは、平文の次のブロックM<sub>2</sub>の入力に対応して、図2 5の暗号化処理E 6 7に出力されることになる。M<sub>2</sub>以下のブロックに対する動作は同様であるので、以下の説明を省略する。

【0026】図2 7 (b)において、暗号文の最初の8ビットC<sub>1</sub>の入力時点で、レジスタ9 3に格納されている初期ベクトルI Vがf関数9 4に与えられ、前述と同様にセクタ9 5、レジスタ9 3を介する4回のループの処理によって1 6段のf関数を用いる変換などが行われ、図2 5の暗号化処理E 7 1の出力としての64ビットのうち左8ビットがE X O R回路9 7に出力され、入力暗号文ブロックC<sub>1</sub>との排他的論理和がとられ、平文M<sub>1</sub>が出力される。

【0027】レジスタ9 3の内容としての初期ベクトルI Vの値がf関数9 4に出力される時点で、同時にその内容は8ビット左シフトされ、5 6ビットがレジスタ9

6に格納される。そして4回目のループの終了時点でその56ビットが左側、入力暗号文8ビットC<sub>1</sub>が右側に配置される形式で、セクタ95を介してレジスタ93に64ビットのデータが格納される。このデータは、次の暗号文ブロックC<sub>2</sub>の入力に対応して、f関数94に与えられるものである。暗号文ブロックC<sub>2</sub>以下に対する復号処理については同様であるので、その説明を省略する。

## 【0028】

【発明が解決しようとする課題】このようなDES暗号化処理を高速化するために、従来は特にf関数を用いた16段の変換処理の高速化を目的として、f関数の変換回路をできるだけ段数多く接続してループ損を削減したり、f関数を含む変換の処理手順を変更して1回のループに対する遅延を削減したり、例えばCBC暗号化処理において操作モードに対応するモード処理を暗号化の処理サイクル、すなわちループの中に入れて処理に必要なクロック数を減らすというような方式が用いられていた。これらの方式に関しては、次の米国特許に開示されている。

【0029】U. S. Patent 5,381,480, Butter et al. "System for Translating Encrypted Data" しながら f 関数を用いる変換をできるだけ段数多く接続するとしても、段数を多くすればするほどハードウェア量が増大し、コストが大きくなるために、あまり段数を多くすることもできないという問題点があった。

【0030】次にDESブロック暗号の操作モードとしては前述のようにECB、CBC、CFB、OFBの4つの操作モードが存在する。ECBモードはDES暗号化処理と復号処理とを組み合わせるだけであり、操作モードのための特別なモード処理回路は不必要である。OFBモードにおいては、DES暗号化の処理ループ内では、前のブロックに対する処理結果を用いてそのまま次のブロックに対する処理を行うことができ、モードに対応する特別な処理としてはDESの最終処理結果と入力ブロックデータとの間で排他的論理和をとるだけでよく、処理ループに対しては遅延の発生を考慮しなくてよい。

【0031】これに対してCBCモードでは、例えば初期ベクトルIVの値と入力ブロックとの排他的論理和をとり、この値に対してDES処理を行う必要がある。またCFBモードでは、暗号化処理において初期ベクトルIVを格納していたレジスタの内容更新時において、初期ベクトルの値の左シフトを行い、右側にDES処理の処理結果の一部と入力ブロックとの排他的論理和の値を設定する必要がある。

【0032】すなわちCBCに対するモード処理はDES処理の最初で、またCFBに対するモード処理は最後で行われるという違いがある。これらの処理は共にDES処理のループの中に含まれている。

【0033】DES暗号化の処理を行う処理回路としては、これらの4つの操作モードの全てを実行できる回路を構成することが望ましいが、従来においては特にCBCモードとCFBモードとに共通に用いられる処理回路を構成することが困難であるという問題点があった。これはこれらの2つのモードに対応するモード処理が一方はDES処理の処理ループの最初、他方は最後に入っていることに起因する。

【００３４】本発明は特にＣＢＣモードとＣＦＢモードの両方を実行できる暗号処理回路を提供することと、その回路における処理遅延をできるだけ削減することを目的とする。

## 【0035】

【課題を解決するための手段】図1は本発明の第1の実施形態に対応する原理構成ブロック図である。同図は、例えば操作モードとしてCBCモードと、CFBモードとの両方のDES暗号化処理を実行できる、ブロック暗号化の処理を行う高速暗号処理回路の構成ブロック図である。

20 【0036】図1において、データ格納手段1はブロック暗号化処理に用いられる初期ベクトル、処理中間値、または処理最終結果を格納するものであり、例えばレジスタである。

【0037】関数処理手段2はブロック暗号化処理、例えばDES処理内で用いられる関数処理としてのf関数を用いる複数段の変換を実行するものである。モード処理手段3はデータ格納手段1と関数処理手段2との間に備えられ、データ格納手段1に格納されているデータを用いてブロック暗号化の操作モード、例えばCBCモードまたはCFBモードに対応する処理を実行するものである。

【0038】CFBモードの処理において、関数処理手段2によって、DES暗号への入力ブロックデータの暗号化における前述の関数処理としての非線形関数を用いる16段の変換が複数回、例えば4回に分けて実行される。その場合には1回の関数処理においては、非線形関数を用いる4段の変換が実行される。

【0039】この複数回の処理の最後の1回を除く毎回の実行結果は、関数処理手段2によってデータ格納手段1に格納される。そしてこの最後の1回の処理結果を用いて入力ブロックデータに対する暗号化ブロックデータを出力するために、暗号化データ出力手段が更に備えられる。

【0040】またこの複数回の処理の最初の回の実行の前に、モード処理手段3によってモード処理が実行され、その実行結果が関数処理手段2に出力されると共に、モード処理の実行結果の一部が格納されるモード処理結果格納手段が更に備えられる。そして関数処理手段2による複数回の処理の最後の回の実行時に、モード処理結果格納手段によって前述のモード処理の実行結果の

一部がデータ格納手段1に出力されると共に、関数処理手段2によって実行結果の一部がデータ格納手段1に出力される。

【0041】CBCモードの暗号化処理の実行時においては、CFBモードにおけると同様に、前述のDES暗号へのブロックデータの暗号化における関数処理としての非線形関数を用いる16段の変換が、複数回に分けて実行される。そしてこの複数回の処理の毎回の実行結果は、関数処理手段2によってデータ格納手段1に格納される。

【0042】また関数処理手段2による複数回の処理の最初の回の実行の前に、モード処理手段3によって入力されるブロックデータとデータ格納手段に格納されているデータとを用いたモード処理が実行され、そのモード処理の結果が関数処理手段2に出力される。

【0043】図2は本発明の第2の実施形態に対応する原理構成ブロック図である。同図においては図1のモード処理手段3に対応する第1のモード処理手段4に加えて、関数処理手段2とデータ格納手段1との間に備えられ、ブロック暗号化の操作モードに対応する第2の処理

【0044】本発明の第2の実施形態においては、第1の実施形態におけると同様に、暗号化回路はCFBモードとCBCモードとの両方の操作モードを実行できるものであるが、CBCモードにおけるモード処理の基本的な部分を実行するのは第1のモード処理手段4であるのに対して、CFBモードにおける基本的なモード処理は第2のモード処理手段5によって実行される。

【0045】CFBモードにおいて、関数処理手段2による非線形関数を用いる16段の変換は、第1の実施形態におけると同様に複数回に分けて実行されるが、その最初の回においてデータ格納手段1に格納されているデータは、第1のモード処理手段3による特別な処理が行われることなく、そのまま関数処理手段2に与えられる。関数処理手段2による複数回の処理の最後の回を除く毎回の処理結果はデータ格納手段1に格納される。

【0046】関数処理手段2による最終回の処理結果を用いて、入力ブロックデータに対する暗号化ブロックデータが第1の実施形態におけると同様に出力されると共に、その処理結果の一部と入力データブロックを用いて第2のモード処理手段によって第2のモード処理が実行され、その結果と、複数回の最初の回の実行の前にデータ格納手段1に格納されていたデータの一部とがデータ格納手段1に格納され、次のブロックデータの暗号化処理のために用いられる。

【0047】CBCモードにおいては、第1のモード処理手段4によって第1の実施形態に対するモード処理手段3によると同様の処理が実行され、第2のモード処理手段は実質的に何らの作用を行うこともなく、結果的に第1の実施形態におけると同様の処理が実行される。

【0048】本発明の暗号処理方法においては、例えばDESブロック暗号化の処理を行う暗号処理方法において、1つのブロックデータの暗号化処理の終了時点で処理結果が出力されると共に、初期ベクトルが格納されていたレジスタにその暗号化処理における処理中間値、または処理最終結果が格納される。

【0049】そしてレジスタに格納されたデータとその1つのブロックデータ、またはその1つのブロックに続く次のブロックデータとの間で、そのブロック暗号化の操作モードに対応する演算が実行される。更にその演算結果が、その1つのブロックに続く次のブロックデータに対する暗号化処理における初期ベクトルに相当する値として用いられ、次のブロックデータに対する暗号化処理が行われる。

【0050】本発明の暗号化処理方法においては、前述のレジスタに最初に初期ベクトルが格納されており、まずその初期ベクトルに対しては例えばCBCモードでは入力ブロックデータとの間で操作モードに対応する演算が実行された後に、CFBモードでは操作モードに対応する特別な演算は実質的に実行されることなく、演算結果に対応して非線形関数を用いる16段の変換が複数回に分けて実行される。この暗号化処理の方法は、前述の第1の実施形態において用いられるものである。

【0051】以上説明したように、本発明によればCBCモードとCFBモードとの両方を実行する暗号化回路を実現することができる。また第1の実施形態においては2つのモードに対応するモード処理手段を共通化することが可能になる。

【0052】

【発明の実施の形態】図3は本発明の第1の実施形態において用いられるCFBモードの暗号化回路の構成を示す。同図は図27(a)に示した従来例に対応するものであるが、CBCモードの暗号化回路と組み合わせて使用するために図27(a)において用いられていたEXOR回路89が省略され、その代わりにレジスタ11の出力側にEXOR回路12が用いられている。

【0053】図3において初期ベクトルIVはレジスタ11に格納されている。平文の最初の8ビットM<sub>1</sub>の入力時点で、レジスタ11から初期ベクトルIVの値がf関数(処理回路)13に与えられる。この時EXOR回路12に対してレジスタ14から、例えば“0”が出力されることにより、レジスタ11に格納されていた初期ベクトルIVの値はそのままf関数13に与えられる。

【0054】f関数13は、図27(a)におけると同様に例えば4段のf関数を用いる変換回路からなり、セクタ15、レジスタ11を介する4回のループの処理によってf関数を用いた16段の変換が行われるものとする。また前述のように初期転置IP、第16段目の結果に対する左右32ビットの入替え、逆転置IP<sup>-1</sup>を含むものとする。

## 11

【0055】4回のループの後のf関数13の出力のうち左8ビットはE XOR回路18に与えられ、入力平文ブロックM<sub>1</sub>との排他的論理和がとられ、最初の暗号文ブロックC<sub>1</sub>として出力される。

【0056】ブロックM<sub>1</sub>の入力に対応してレジスタ11から初期ベクトルIVが出力される時点で、その内容のうち、左側56ビットはそのままレジスタ11から、また右側8ビットはE XOR回路12を介して与えられる形式でレジスタ17側に出力されるが、その64ビットは8ビットシフトされて最も左側の8ビットが捨てられ、レジスタ17に格納される。そして4回のループの終了時点でその内容の56ビットが左側に、f関数13の出力のうちの左8ビットが右側に格納される形式で、セクタ15を介してレジスタ11に64ビットのデータが格納される。またこの時点でレジスタ14には入力ブロックM<sub>1</sub>の8ビットが格納される。ここでレジスタ11に格納されている64ビットは、そのままでは次の入力ブロックM<sub>2</sub>の入力時点でf関数13に対してDES暗号化処理の対象として用いることはできない。

【0057】ブロックM<sub>2</sub>の入力時点で、レジスタ11の格納内容のうち右側8ビットはE XOR回路12に出力され、レジスタ14に格納されている最初のブロックM<sub>1</sub>の8ビットとの排他的論理和がとられ、レジスタ11からそのまま出力される左側56ビットと共にf関数13に入力される。この時実質的にf関数13への入力となる64ビットは、前述と同様に左に8ビットシフトされ、56ビットがレジスタ17に格納される。これは次のブロックM<sub>3</sub>の入力時点におけるf関数13への入力データ作成のためである。

【0058】f関数13、セクタ15、レジスタ11を介する4回のループの終了時点で、f関数13の出力のうち左8ビットと入力M<sub>2</sub>の8ビットとの排他的論理和がE XOR回路18によってとられ、暗号文系列の次のブロックC<sub>2</sub>として出力される。この4回のループ処理の間はレジスタ14の格納内容は出力されず、E XOR回路12には例えば“0”が与えられるものとする。以下同様の処理が行われるため、その説明を省略する。

【0059】図4は本発明の第1の実施形態としての暗号処理回路の構成ブロック図である。同図はCFBモードとCBCモードとの両方のモードの処理を実行する暗号処理回路の構成ブロック図である。この回路を用いたCFBモード暗号化動作について、図5～図9を用いて説明する。

【0060】図5はCFBモード暗号化動作（その1）の説明図である。ここで使用される回路の結線部分については実線で、また使用されない部分については破線で示してある。まず最初の平文ブロックM<sub>1</sub>の入力（①から）に対応して、レジスタ21に格納されている初期ベクトルIVがセクタ22、23、E XOR回路24、25を介してf関数26に与えらる。この時セクタ2

## 12

2、23はレジスタ21からの出力を選択し、またE XOR回路24、25に対してはセクタ31、32を介して56ビットと8ビットの“0”のデータが入力されている。このためレジスタ21に格納されていた初期ベクトルIVの値はそのままf関数26に与えられる。またレジスタ21に格納されていたIVの値は8ビットシフトされるだけで、実質的にそのままレジスタ28に格納される。そしてf関数26に含まれる4段の変換回路の処理結果はセクタ27を介してレジスタ21に格納される。

【0061】図6はCFBモード暗号化動作（その2）の説明図である。同図はf関数26、セクタ27、レジスタ21を介する4回のループのうち、第2回目と第3回目のループにおける処理動作の説明図である。これらのループにおいてはセクタ22、23はレジスタ21からの出力を選択し、またセクタ31、32は56ビットの“0”と8ビットの“0”をそれぞれE XOR回路24、25に出力しているため、レジスタ21のデータはそのままf関数26に与えられ、4段の処理結果はセクタ27を介してレジスタ21に格納される。

【0062】図7はCFBモード暗号化動作（その3）の説明図である。同図において、レジスタ21に格納されている3回目のループの処理結果は、前述と同様にセクタ22、23、E XOR回路24、25を介してそのままf関数26に与えられる。f関数26の出力は16段のf関数を用いた初期ベクトルIVの暗号化処理結果であるため、その左8ビットが取り出され、入力①に与えられる平文の最初のブロックM<sub>1</sub>の8ビットとの論理和がE XOR回路30によってとられ、暗号文の最初のブロックC<sub>1</sub>として出力①から出力される。また平文ブロックM<sub>1</sub>の8ビットはレジスタ33に格納される。

【0063】同時にレジスタ28に格納されていた56ビットが左側に、f関数26の出力のうち左8ビットが右側に格納される形式で、セクタ27を介してレジスタ21に64ビットのデータが格納される。

【0064】図8はCFBモード暗号化動作（その4）の説明図である。ここでは入力の平文ブロックM<sub>2</sub>に対応する最初の処理が実行される。まずレジスタ21に（その3）で格納された64ビットのデータが、セクタ22、23を介して2つのE XOR回路24、25に与えられる。この時セクタ31は56ビットの“0”を、32はレジスタ33に格納されているデータ、すなわち入力平文の最初のブロックM<sub>1</sub>のデータを出力しており、E XOR回路24は64ビットのうち左側56ビットをそのままf関数26に出力するが、回路25は右側8ビットとM<sub>1</sub>の8ビットの排他的論理和を出力することになり、その結果の64ビットがブロックM<sub>2</sub>の入力に対応する4回のループの処理の最初にf関数26に与えられる。また実質的にこの入力は8ビットシフトされ、レジスタ28に格納される。そしてf関数26にお

10

20

30

40

50

ける4段の変換処理(初期転置を含む)の結果はセレクト27を介してレジスタ21に格納される。

【0065】図9はCFBモード暗号化動作のタイミングチャートである。動作フローに対する数字は、図5～図8で説明した暗号化処理その1～その4に対応する数字である。平文の最初のブロックM<sub>1</sub>の入力に対しては、4つのクロックに対応してその1、その2、その2、その3の処理が実行されて、出力の最初のブロックC<sub>1</sub>が得られる。次の入力ブロックM<sub>2</sub>以下に対しては、4つのクロックに対応してその4、その2、その2、その3の処理が実行され、出力ブロックC<sub>2</sub>以下が得られる。なお、例えば暗号文ブロックC<sub>1</sub>は図7で説明したように、4回目のループの最後にf関数26の出力と平文ブロックM<sub>1</sub>との排他的論理和の結果として短時間出力されるが、図にはその出力期間を約1クロック分示してある。出力端子①の前にラッチ回路を設ければこの期間を次の暗号文ブロック出力の前まで延長することも可能である。

【0066】図10はCFBモード復号動作(その1)の説明図である。同図においては、暗号文の最初のブロックC<sub>1</sub>の入力に対応して、レジスタ21に格納されている初期ベクトルIVの値がセクタ22、23、EXOR回路24、25を介してそのままf関数26に与えられる。また初期ベクトルIVは8ビット左シフトされるだけでレジスタ28に格納される。そして初期転置を含むf関数による4段の処理の終了後、その結果はセクタ27を介してレジスタ21に格納される。

【0067】図11はCFBモード復号動作(その2)の説明図である。同図は、f関数を用いる処理としての4回のループのうちの2回目のループと3回目のループの処理動作の説明図である。レジスタ21に格納されている64ビットのデータはセクタ22、23、EXOR回路24、25を介してf関数26に与えられ、f関数26の出力はセクタ27を介してレジスタ21に格納される。

【0068】図12はCFBモード復号動作(その3)の説明図である。同図はf関数を用いる4回目のループの処理動作を示す。レジスタ21に格納されている64ビットのデータは、セクタ22、23、EXOR回路24、25を介してf関数26に与えられる。f関数を用いる4段の変換、左右32ビットの入替え、および逆転置IP<sup>-1</sup>の処理の後に、その処理結果としての64ビットのうち左8ビットがEXOR回路30に与えられ、入力①から入力される最初の暗号文ブロックC<sub>1</sub>の8ビットとの排他的論理和がとられ、その結果は出力①から平文の最初のブロックM<sub>1</sub>として出力される。また同時にレジスタ28に格納されている56ビットが左側に、入力暗号文ブロックC<sub>1</sub>の8ビットが右側に配置される形式で、セクタ27を介してレジスタ21に64ビットのデータとして格納される。

【0069】図13はCFBモード復号動作のタイミングチャートである。動作フローにおける数字は図10～図12としてのその1～その3の処理を示す。入力暗号文の最初のブロックC<sub>1</sub>の入力に対しては、4つのクロックにおいてその1、その2、その2、その3の処理が実行され、出力平文の最初のブロックM<sub>1</sub>が出力される。次の入力暗号文ブロックC<sub>2</sub>以下の入力に対しても、全く同様の動作が実行される。

【0070】図14はCBCモード暗号化動作(その1)の説明図である。CBCモードにおいては入力平文ブロックは入力②から入力され、出力暗号文ブロックは出力②から出力される。まず入力②から入力された最初の平文ブロックM<sub>1</sub> 64ビットは、そのうち左56ビットがセクタ31に、右8ビットが32に与えられる。そしてセクタ31はその56ビットをEXOR回路24に、32は8ビットを25に与える。

【0071】一方レジスタ21に格納されている初期ベクトルIVはセクタ22、23を介してEXOR回路24、25に与えられ、初期ベクトルIVと最初の平文入力ブロックM<sub>1</sub>との排他的論理和がとられ、その結果はf関数26に与えられる。そして初期転置IPを含む4段のf関数を用いる変換の結果は、セクタ27を介してレジスタ21に格納される。

【0072】図15はCBCモード暗号化動作(その2)の説明図である。同図はf関数を用いる4回のループのうち、2回目および3回目のループにおける動作の説明図である。レジスタ21に格納されている64ビットのデータは、セクタ22、23を介して2つのEXOR回路24、25に与えられる。この時セクタ31、32はそれぞれ56ビットの“0”、8ビットの“0”を出力しているため、その64ビットのデータはそのままf関数26に与えられる。そしてf関数を用いる4段の変換処理の結果は、セクタ27を介してレジスタ21に格納される。

【0073】図16はCBCモード暗号化動作(その3)の説明図である。同図はf関数を用いる4回目のループの動作を説明するものである。レジスタ21に格納されている64ビットのデータは、前述と同様にセクタ22、23、EXOR回路24、25を介して、そのままf関数26に与えられる。f関数を用いる4段の変換、左右32ビットの入れ替え、および逆転置IP<sup>-1</sup>の処理結果は、セクタ27を介してレジスタ21に格納される。レジスタ21に格納された64ビットのデータは、EXOR回路34を介してそのまま出力②から出力暗号文の最初のブロックC<sub>1</sub>として出力される。この時セクタ35は64ビットの“0”をEXOR回路34に出力している。またレジスタ21に格納された出力暗号文の最初のブロックC<sub>1</sub>は、次の平文ブロックM<sub>2</sub>の入力時点で暗号化動作(その1)によって入力ブロックM<sub>2</sub>と、前述と同様にして、排他的論理和がとられること



になる。

【0074】図17はCBCモード暗号化動作のタイミングチャートである。入力平文ブロックM<sub>1</sub>の入力に対応して、4つのクロックにおいてその1、その2、その2、その3の動作が実行され、出力暗号文の最初のブロックC<sub>1</sub>が出力される。平文ブロックM<sub>2</sub>以下の入力に対する動作は全く同様である。

【0075】図18はCBCモード復号動作（その1）の説明図である。CBCモード復号動作では、初期ベクトルIVはレジスタ37に格納されている。最初の暗号文ブロックC<sub>1</sub>が入力②から入力されると、セクタ31ではその左側56ビット、32では右側8ビットが選択されて、2つのEXOR回路24、25に与えられる。この時セクタ22、23は共に56ビットの“0”、8ビットの“0”を出力しており、このため入力暗号文ブロックC<sub>1</sub>の64ビットはそのままf関数26に与えられる。そして初期転置IPを含む、f関数を用いる4段の処理結果は、セクタ27を介してレジスタ21に格納される。

【0076】図19はCBCモード復号動作（その2）の説明図である。同図はf関数を用いる4回のループのうち2回目、3回目のループにおける動作説明図である。同図においてレジスタ21に格納されている64ビットのデータは、セクタ22、23、EXOR回路24、25を介してそのままf関数26に与えられ、f関数を用いる4段の変換の処理結果はセクタ27を介してレジスタ21に格納される。またレジスタ37に格納されている初期ベクトルIVは例えば3回目（2回目でもよい）のループの処理時にレジスタ36に格納される。

【0077】図20はCBCモード復号動作（その3）の説明図である。同図において4回目のループとして、レジスタ21に格納されている64ビットのデータはそのままf関数26に与えられ、f関数を用いる4段の変換、左右32ビットの入れ替え、および逆転置IP<sup>-1</sup>の処理結果がセクタ27を介してレジスタ21に与えられる。レジスタ21に格納された64ビットのデータはEXOR回路34に与えられ、レジスタ36に格納されている初期ベクトルIVとの排他的論理和がとられ、出力②から出力平文の最初のブロックM<sub>1</sub>として出力される。またこの時次の暗号文ブロックC<sub>2</sub>に対する（その3）の動作においてEXOR回路34に出力するためのデータとして、入力暗号文の最初のブロックC<sub>1</sub>がレジスタ37に格納される。

【0078】図21はCBCモード復号動作のタイミングチャートである。同図において暗号文の最初のブロックC<sub>1</sub>の入力に対しては、4つのクロックにおいてその1、その2、その2、その3の動作が行われ、平文の最初のブロックM<sub>1</sub>が出力される。以下暗号文の次のブロックC<sub>2</sub>・・・の入力に対する動作は全く同様である。

【0079】図22は本発明の第2の実施形態としてのCFBモードとCBCモードの2つの動作を実行する暗号処理回路の構成ブロック図である。図4の第1の実施形態においては、2つのEXOR回路24、25がCBCモードだけでなく、CFBモードの暗号化動作においても、f関数26に与えるべき64ビットのデータのうちで左側8ビットに対する排他的論理和をとるために用いられていたが、図22においてはこの排他的論理和がEXOR回路45によってとられる点に基本的な相違がある。このため図4のセクタ22、23は1つのセクタ41、EXOR回路24、25は1つのEXOR回路42、セクタ31、32は1つのセクタ43によって実現されている。

【0080】図22の第2の実施形態と図4の第1の実施形態との相違は基本的にCFBモード暗号化動作にあるため、その動作を中心に図22について説明する。図22では、前述のCFBモード暗号化動作（その1）に対応して、レジスタ21に格納されている初期ベクトルIVはセクタ41、EXOR回路42を介してそのままf関数26に与えられ、その処理結果はセクタ46を介してレジスタ21に格納される。暗号化動作（その2）における動作も同様である。

【0081】暗号化動作（その3）に相当して、4回目のループとしてのf関数26の出力のうち左8ビットが、入力①から入力される最初の平文ブロックM<sub>1</sub>とEXOR回路30によって排他的論理和がとられ、出力①から最初の暗号文ブロックC<sub>1</sub>として出力される。また同時にレジスタ44に格納されている56ビットが左側に、f関数26の出力のうち左8ビットと入力ブロックM<sub>1</sub> 8ビットとの排他的論理和がEXOR回路45によってとられた結果が右側に配置される形式で、セクタ46を介して64ビットのデータがレジスタ21に格納される。このレジスタ21に格納されたデータのうち右8ビットはすでに入力との排他的論理和がとられたものであり、次の平文ブロックM<sub>2</sub>の入力に対応してセクタ41、EXOR回路42を介してそのままf関数26に与えられることになる。

【0082】CFBモード復号、CBCモード暗号化、およびCBCモード復号の動作は第1の実施形態におけると実質的に同じであるため、その説明を省略する。図23は、第1の実施形態と第2の実施形態とにおいて、1つのクロックに含まれる処理の説明図である。図4に示した第1の実施形態および図22に示した第2の実施形態のそれぞれにおいて、f関数26を用いた16段の変換は4段ずつ4つのクロックに分けて行われ、1つのブロックデータに対する暗号化処理のために4つのクロックを必要とする点は、いずれの実施形態においても同様である。しかしながら第1の実施形態においては、1つのクロック内で行われる処理の遅延時間の合計が第2の実施形態におけるよりも短くなり、結果的にクロック

の速度を上げることが可能となり、暗号化処理を高速化することができる。

【0083】図23のf関数による1～4段目の変換を含む1回目のループにおいて、下段の第1の実施形態に対応する図4ではレジスタ21からの出力による遅延、セクタ22、23による遅延、E XOR回路24、25による処理の遅延、4段のf関数を用いる変換（初期転置を含む）の遅延、セクタ27による遅延に加えて、レジスタ21におけるデータのセットアップによる遅延が1つのクロック内に含まれることになる。

【0084】それに対して上段の第2の実施形態に対応する図22の回路においては、レジスタ21によるデータ出力の遅延、セクタ41による遅延、E XOR回路42による遅延、f関数を用いる4段の変換による遅延の後に、E XOR回路45による遅延、セクタ27による遅延、レジスタ21のデータセットアップによる遅延が含まれる。すなわち第1の実施形態に比べてE XOR回路45による遅延が余計に含まれ、第2の実施形態においてはその分だけ第1の実施形態に比べてクロックの周期が長くなり、高速化の面では第1の実施形態より劣ることになる。しかしながらこの点を除けば、第2の実施形態においてもCFBモードとCBCモードとの両方の操作モードを実行する暗号処理回路が実現される。

【0085】

【発明の効果】以上詳細に説明したように、本発明によればDES暗号のCBCモードとCFBモードとの両方の操作モードを実行できる暗号処理回路を実現することが可能となる。また第1の実施形態においては、CFBモードにおけるモード処理を次のブロックデータに対する処理の最初に移動することによって、CBCモードの処理を実行する部分と共通の部分を使用することが可能となり、処理ループの最大遅延を短縮することができ、高速処理が可能となる。また同時に排他的論理和の回路の削減も可能となり、ハードウェア量を減少することができ、暗号処理方式の実用性向上に寄与するところが大きい。

【図面の簡単な説明】

【図1】第1の実施形態に対応する原理構成ブロック図である。

【図2】第2の実施形態に対応する原理構成ブロック図である。

【図3】本発明の第1の実施形態において使用されるCFBモード暗号化回路の構成を示すブロック図である。

【図4】本発明の第1の実施形態としての暗号処理回路の構成を示すブロック図である。

【図5】第1の実施形態におけるCFBモード暗号化動作（その1）の説明図である。

【図6】第1の実施形態におけるCFBモード暗号化動作（その2）の説明図である。

【図7】第1の実施形態におけるCFBモード暗号化動

作（その3）の説明図である。

【図8】第1の実施形態におけるCFBモード暗号化動作（その4）の説明図である。

【図9】CFBモード暗号化動作のタイミングチャートである。

【図10】第1の実施形態におけるCFBモード復号動作（その1）の説明図である。

【図11】第1の実施形態におけるCFBモード復号動作（その2）の説明図である。

【図12】第1の実施形態におけるCFBモード復号動作（その3）の説明図である。

【図13】CFBモード復号動作のタイミングチャートである。

【図14】第1の実施形態におけるCBCモード暗号化動作（その1）の説明図である。

【図15】第1の実施形態におけるCBCモード暗号化動作（その2）の説明図である。

【図16】第1の実施形態におけるCBCモード暗号化動作（その3）の説明図である。

【図17】CBCモード暗号化動作のタイミングチャートである。

【図18】第1の実施形態におけるCBCモード復号動作（その1）の説明図である。

【図19】第1の実施形態におけるCBCモード復号動作（その2）の説明図である。

【図20】第1の実施形態におけるCBCモード復号動作（その3）の説明図である。

【図21】CBCモード復号動作のタイミングチャートである。

【図22】本発明の第2の実施形態としての暗号処理回路の構成を示すブロック図である。

【図23】第1の実施形態と第2の実施形態における1つのクロック内で行われる処理の比較を示す図である。

【図24】CBCモードにおける暗号化処理の基本説明図である。

【図25】CFBモードの暗号化処理の基本説明図である。

【図26】CBCモードの暗号処理回路の従来例を示す図である。

【図27】CFBモードの暗号処理回路の従来例を示す図である。

【符号の説明】

1 データ格納手段

2 関数処理手段

3 モード処理手段

4 第1のモード処理手段

5 第2のモード処理手段

21, 28, 33, 36, 37, 44 レジスタ

22, 23, 27, 31, 32, 35, 41, 46 セ

クタ

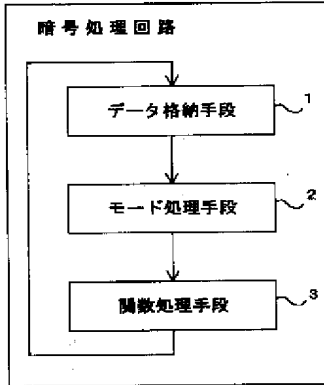
【図1】

【図2】

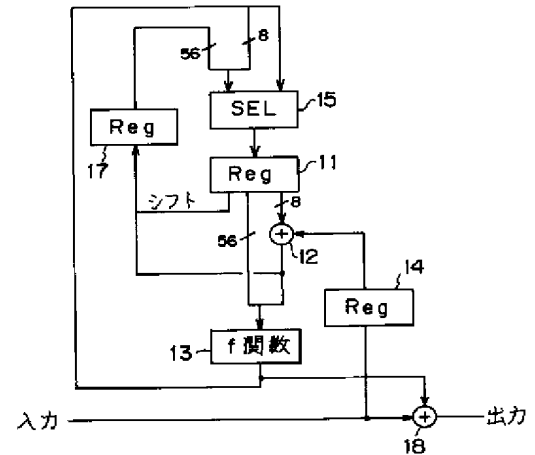
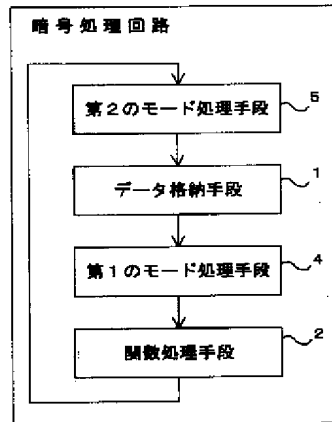
【図3】

本発明の第1の実施形態において使用される  
CFBモード暗号化回路の構成を示すブロック図

第1の実施形態に対応する  
原理構成ブロック図

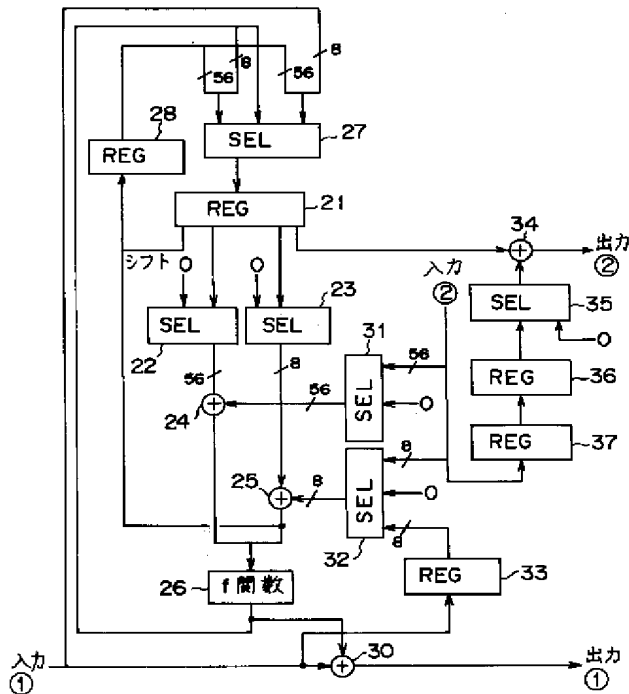


第2の実施形態に対応する  
原理構成ブロック図



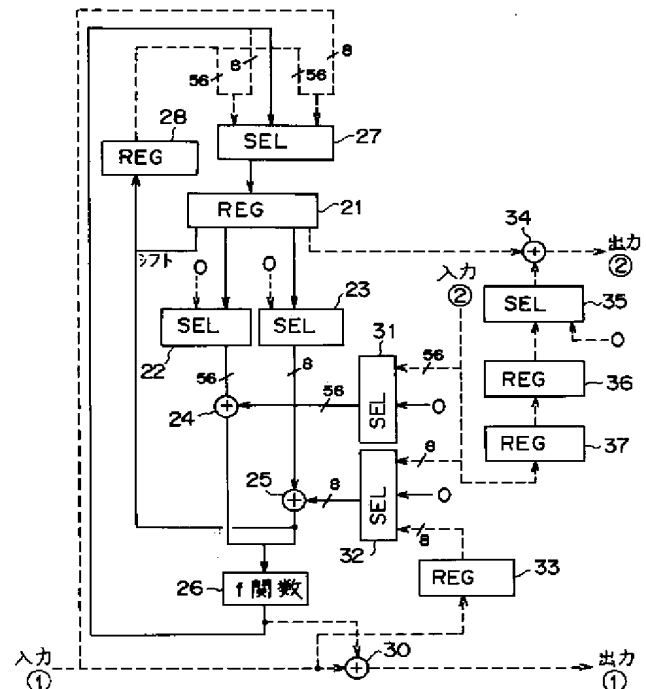
【図4】

本発明の第1の実施形態としての  
暗号処理回路の構成を示すブロック図

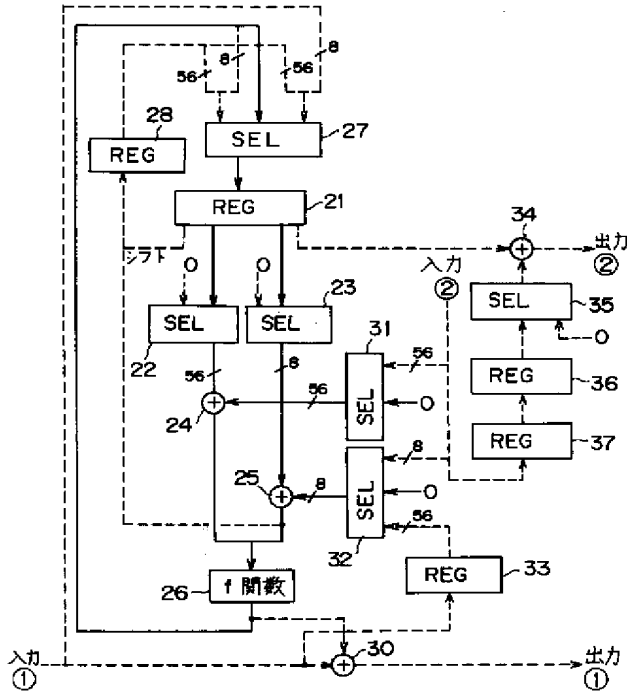


【図5】

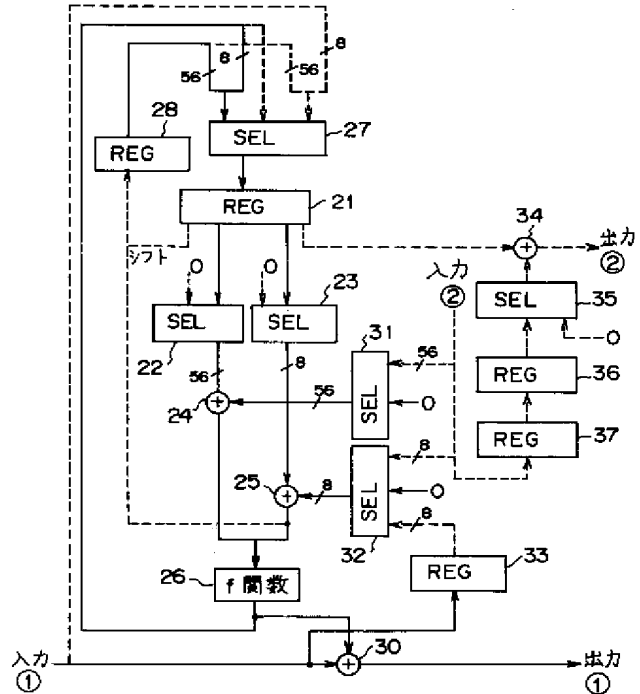
第1の実施形態におけるCFBモード暗号化動作  
(その1)の説明図



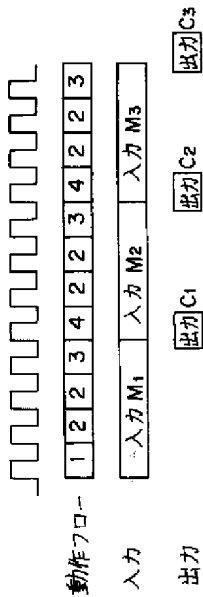
【図6】

第1の実施形態におけるCFBモード暗号化動作  
(その2)の説明図

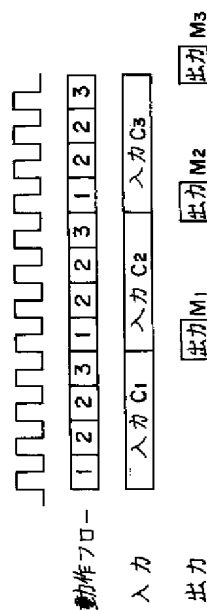
【図7】

第1の実施形態におけるCFBモード暗号化動作  
(その3)の説明図

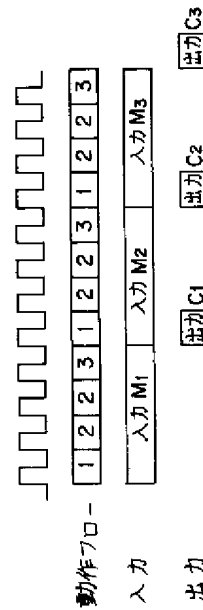
【図9】

CFBモード暗号化動作の  
タイミングチャート

【図13】

CFBモード復号動作の  
タイミングチャート

【図17】

CBCモード暗号化動作の  
タイミングチャート

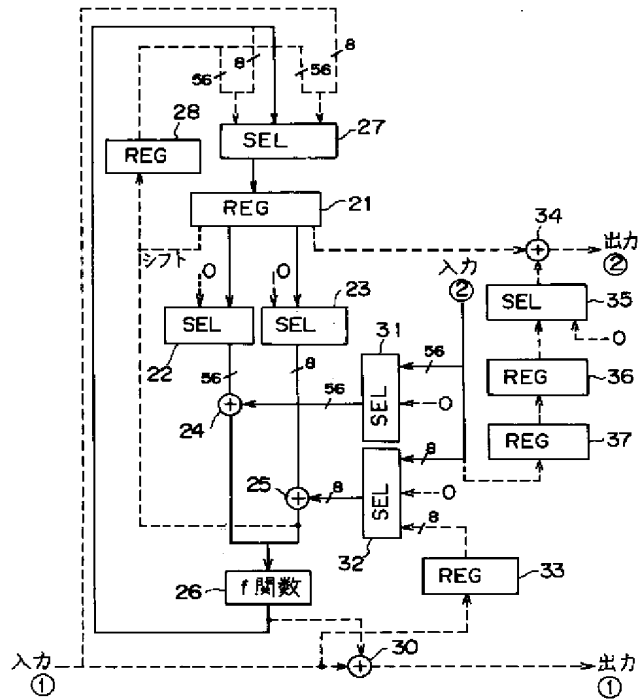




【図14】

第1の実施形態におけるCBCモード暗号化動作

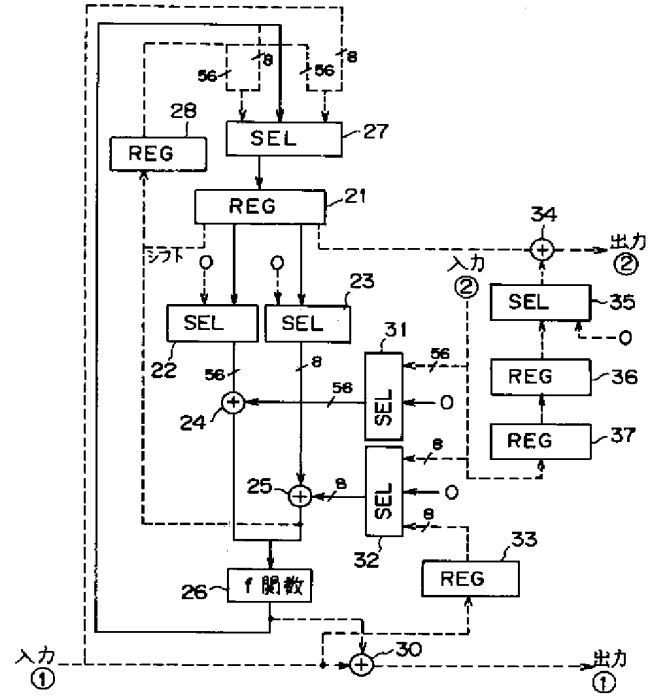
(その1)の説明図



【図15】

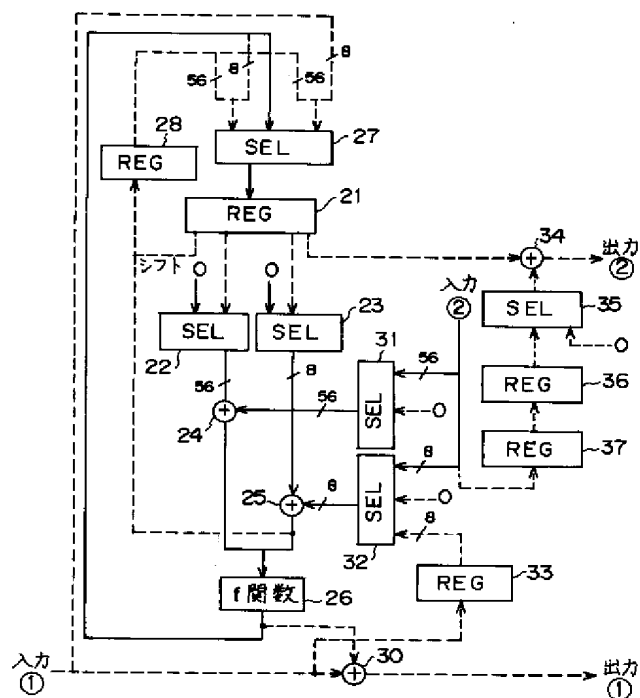
第1の実施形態におけるCBCモード暗号化動作

(その2)の説明図



【例 18】

第1の実施形態におけるCBCモード復号動作  
(その1)の説明図

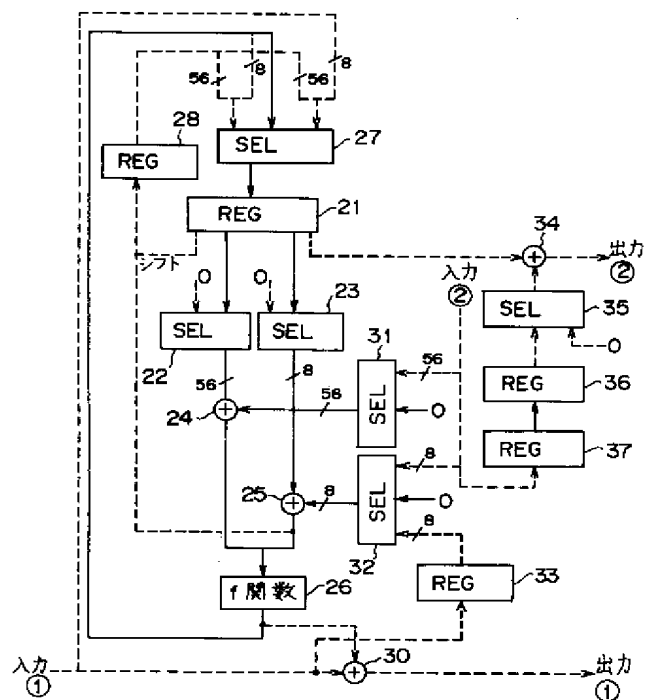




【图 19】

第1の実施形態におけるCBCモード復号動作

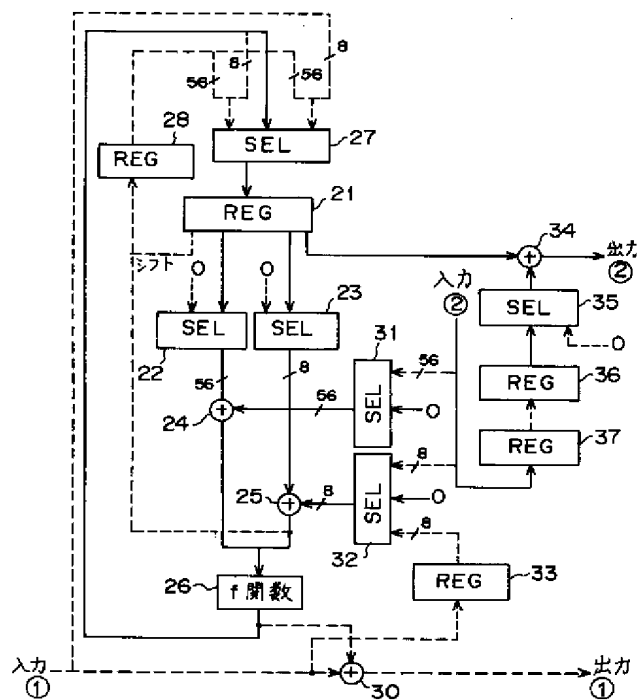
(その2)の説明図



【図20】

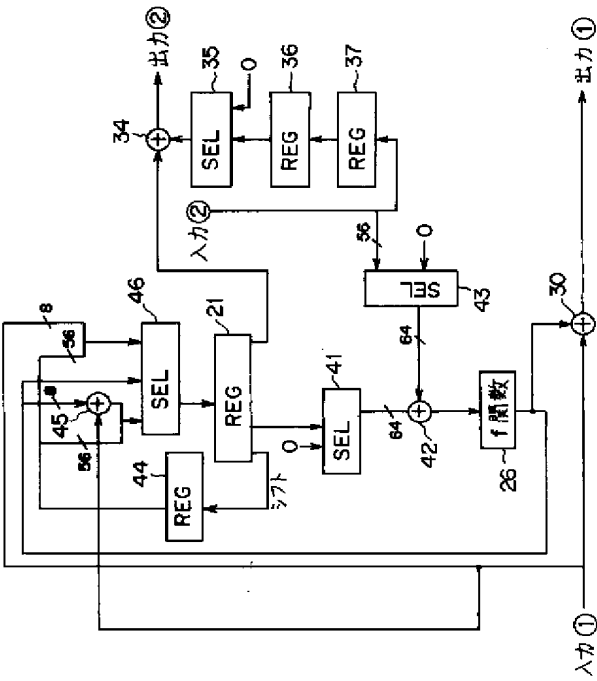
### 第1の実施形態におけるCBCモード復号動作

(その3)の説明図



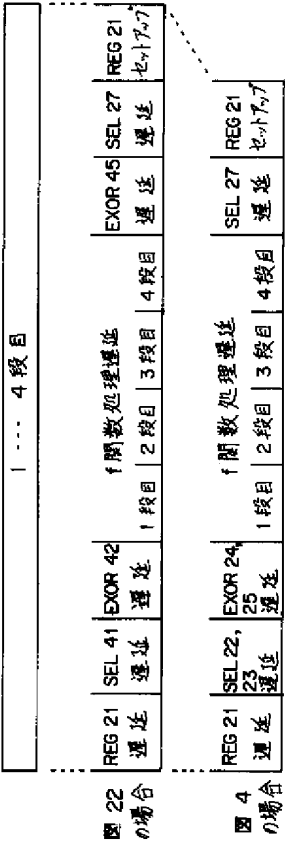
【図22】

本発明の第2の実施形態としての  
暗号処理回路の構成を示すブロック図



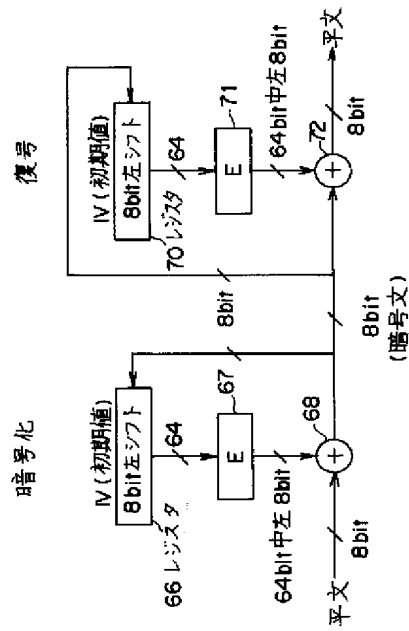
【図23】

第1の実施形態と第2の実施形態における  
1つのクロック内で行われる処理の比較を示す図



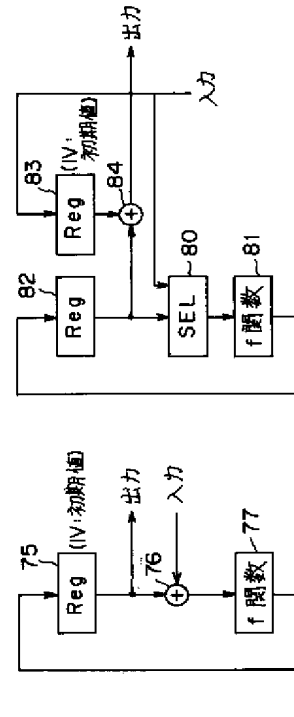
【図25】

CFBモードの暗号化処理の基本説明図



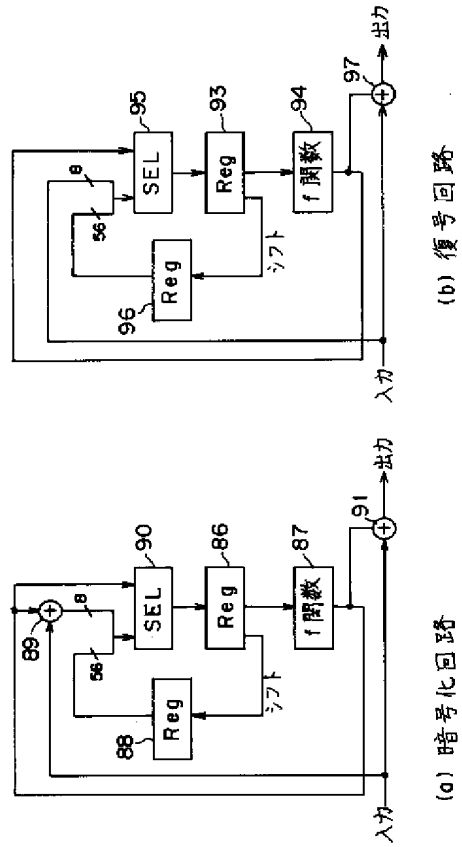
【図26】

CBCモードの暗号処理回路の従来例を示す図



【図27】

CFBモードの暗号処理回路の従来例を示す図



**Disclaimer:**

This English translation is produced by machine translation and may contain errors. The JPO, the INPIT, and those who drafted this document in the original language are not responsible for the result of the translation.

**Notes:**

1. Untranslatable words are replaced with asterisks (\*\*\*).
2. Texts in the figures are not translated and shown as it is.

Translated: 23:48:49 JST 05/13/2008

Dictionary: Last updated 04/11/2008 / Priority: 1. Information communication technology (ICT) / 2. Electronic engineering / 3. Mathematics/Physics

---

## FULL CONTENTS

---

### [Claim(s)]

[Claim 1] The initial vector used for block encryption processing in the cipher-processing circuit which processes block encryption, a processing mean value, or a data storage means to store the processing final result, The high-speed cipher-processing circuit characterized by having between a function processing means to perform function processing used within block encryption processing, and this data storage means and a function processing means, and having a mode processing means to perform processing corresponding to the operation mode of block encryption.

[Claim 2] The high-speed cipher-processing circuit according to claim 1 characterized by said mode processing means being constituted by the exclusive "or" circuit.

[Claim 3] Claim 1 characterized by for said block cipher being a DES code and said operation mode being CFB mode, or a high-speed cipher-processing circuit given in two.

[Claim 4] The high-speed cipher-processing circuit according to claim 3 characterized by for said function processing means dividing 16 steps of conversion using the nonlinear function as said function processing in encryption of the block data to said DES code into multiple times, and performing it.

[Claim 5] The high-speed cipher-processing circuit according to claim 4 characterized by said function processing means outputting the execution result at each time except the last 1 time of said multiple times to said data storage means.

[Claim 6] The high-speed cipher-processing circuit according to claim 4 characterized by having further an encryption data output means by said function processing means to output the encryption block data to input block data using the processing result of the last round of said multiple times.

[Claim 7] While said mode processing means performs mode processing and outputs this execution result to this function processing means before execution of the time of the

beginning of said multiple times by said function processing means The high-speed cipher-processing circuit according to claim 4 characterized by having further a mode processing result storing means to store a part of execution result of this mode processing.

[Claim 8] The high-speed cipher-processing circuit according to claim 7 characterized by this function processing means outputting this a part of execution result to this data storage means while outputting the data which said mode processing result storing means holds to said data storage means at the time of execution of the time of the last of said multiple times by said function processing means.

[Claim 9] Claim 1 characterized by for said block cipher being a DES code and said operation mode being the CBC mode, or a high-speed cipher-processing circuit given in two.

[Claim 10] The high-speed cipher-processing circuit according to claim 9 characterized by for said function processing means dividing 16 steps of conversion using the nonlinear function as said function processing in encryption of the block data to said DES code into multiple times, and performing it.

[Claim 11] The high-speed cipher-processing circuit according to claim 9 characterized by said function processing means outputting the execution result of each time of said multiple times to said data storage means.

[Claim 12] Before execution of the time of the beginning of said multiple times by said function processing means, mode processing is performed using the block data into which said mode processing means is inputted, and the data stored in said storing means. The high-speed cipher-processing circuit according to claim 9 characterized by outputting the result of this mode processing to this function processing means.

[Claim 13] The initial vector used for block encryption processing in the cipher-processing circuit which processes block encryption, a processing mean value, or a data storage means to store the processing final result, A function processing means to perform function processing used within block encryption processing, The 1st mode processing means which it has between this data storage means and function processing, and performs 1st processing corresponding to the operation mode of block encryption, The high-speed cipher-processing circuit characterized by having between this function processing means and a data storage means, and having the 2nd mode processing means which performs 2nd processing corresponding to the operation mode of block encryption.

[Claim 14] While being at the end time of encryption processing of one block data and outputting a processing result in the cipher-processing method of processing block encryption A processing mean value [ in / to the register which stored the initial vector / this encryption processing ], or -- storing the processing final result -- this -- one block data -- or -- this -- [ the operation corresponding to the operation mode of this block encryption / perform and ] between the data of a block of the degree following one block, and the data stored in the

register this result of an operation -- this -- using as a value equivalent to said initial vector in the encryption processing to the following block data following one block -- this -- the cipher-processing method characterized by performing encryption processing to the following block data.

[Claim 15] It is the cipher-processing method according to claim 14 characterized by for said block cipher being a DES code and said operation mode being the CBC mode in CFB mode and \*\*\*\*.

---

#### [Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the high-speed cipher-processing circuit which performs processing in the CBC mode of the DES code as a still more detailed typical block cipher, and CFB mode about the cipher system of digital data.

[0002]

[Description of the Prior Art] Encryption processing of digital data attracts attention in recent years, and the data which is the target of encryption processing is also increasing. For example, the demand of encryption processing of dynamic image data and transaction processing in a server has come out, and the encryption processing itself is asked for improvement in the speed.

[0003] There are a block cipher system which performs 64 bits as a cipher system of digital data, and enciphers for every block, for example as one block, and a stream cipher system which enciphers 1 bit at a time sequentially, for example.

[0004] The target cipher system [ this invention ] is a block cipher, and is the most typical DES code also in it. DES (data encryption standard) is cryptographic algorithm which is the data code standard defined by Standards Division, the U.S. Department of Commerce, and is used in January, 1977 present most extensively.

[0005] To the DES code, four modes are defined as the operation mode. The mode is four, ECB (electronic code block), CBC (cipher block chaining), CFB (cipher feedback), and OFB (output feed back).

[0006] Since it is two modes, CBC and CFB, it explains that direct relation is in this invention among these four modes about processing of the encryption and the decode in these two operation modes using drawing 24 - drawing 27 .

[0007] Drawing 24 is the basic explanatory view of the encryption and the decode in the CBC mode. Although the plaintext sequence M1 divided into 64 bits, M2, M3, and ... are inputted in this mode first -- M1 as 64 bits of the beginning receive, the exclusive OR 51 with the initial

vector IV is taken, and encryption processing E52 is performed to the result -- the cipher sequence C1, C2, C3, and C1 as 64 bits of the beginning of ... It is obtained. E52 as encryption processing contains the initial transposition IP, 16 steps of conversion of the same structure which used f function, exchange of 32 bits of right and left, and reverse transposition IP-1 here.

[0008] M2 as following 64 bits To an input, it is C1 as first 64 bits of a cipher. The exclusive OR 56 is taken, encryption processing E57 is performed to the result, and it is C2 as 64 bits next to a cipher. \*\*\*\*\*. 64 bits of ciphers are created at a time like the following.

[0009] In decoding processing, it is C1 as first 64 bits of a cipher. Decoding processing D53 is performed to an input. Although this processing contains the initial transposition IP, 16 steps of conversion using f function, exchange of 32 bits of right and left, and reverse transposition IP-1 like the encryption processing E52, unlike encryption processing, in 16 steps of conversion, it is used in the sequence that a key sequence is reverse. To the result of the decoding processing D53, the exclusive OR 54 with the initial vector IV is taken, and it is M1 as first 64 bits of a plaintext. It is obtained.

[0010] C2 as 64 bits next to a cipher After decoding processing D58 was performed to the input, it is C1 as first 64 bits of a cipher to the result. The exclusive OR 59 is taken and it is M2 as 64 bits next to a plaintext. It is obtained. The plaintext as a decoded result is created like the following.

[0011] Drawing 25 is the basic explanatory view of operation in CFB mode. CFB mode performs random-number-generation operation, a plaintext is divided into every k bits which is 64 or less [ 1 or more ], and the sequence is similarly made into M1, M2, M3, and ... in drawing 24 .

[0012] M1 as a plaintext, for example, 8 bits of the beginning, Corresponding to an input, encryption processing E67 of the initial vector IV stored in the register 66 is performed. This encryption processing itself is the same also in the CBC mode of drawing 24 . only eight of 64 bits as a result of encryption processing of left-hand side are taken out -- M1 the exclusive OR 68 -- it can take -- 8 bit C1 of the beginning of a cipher while being outputted -- 8 bit C1 as the cipher It is stored in the rightmost side of a register 66. That is, it acts as a 8-bit left shift, 8 bits of left-hand side are thrown away, and the initial vector IV stored in the register 66 is C1 most to right-hand side. It will be stored.

[0013] The following 8 bit M2 of a plaintext Corresponding to an input, encryption processing E67 to the contents of storing of a register 66 is performed. 8 bits of lefts of the 64 bits of the result, and M2 the exclusive OR 68 -- it can take -- the following 8 bit C2 of a cipher \*\*\*\*\* -- while being outputted, the 8 bits are stored in the rightmost side of a register 66. That is, it acts as a 8-bit left shift, 8 more bits of the initial vector IV are thrown away, and the contents of storing of a register 66 are C2 to the rightmost side of a register 66. It will be stored. The



following 8 bit C3 of a cipher sequence About the following, same operation is completely performed.

[0014] In the decoding processing in CFB mode, it is C1 the first 8 bits of a cipher first. Corresponding to an input, encryption processing E71 of the initial vector IV stored in the register 70 is performed. Unlike the CBC mode of drawing 24 , also in decoding processing, not the processing D but the processing E is used. 8 bits of lefts are taken out among 64 bits as a result of this processing, and it is C1 of a cipher. The exclusive OR 72 is taken and it is M1 the first 8 bits of a plaintext. It is outputted. In encryption processing, it acts as a 8-bit left shift of the contents of the register 70 similarly simultaneously with it, and a cipher is C1 8 bits most to right-hand side. It is stored.

[0015] The following 8 bit C2 of a cipher sequence The processing corresponding to an input is completely the same. That is, encryption processing E71 is performed to the contents of the register 70, and it is 8 bits of lefts in 64 bits of the result, and C2. The exclusive OR 72 is taken and the degree of a plaintext is M2 8 bits. while being obtained As for the contents of the register 70, it acts as a 8-bit left shift, and a cipher is C2 8 bits. It is stored in the rightmost side of a register 70. The cipher sequence C3 and decoding processing of ... are performed like the following.

[0016] Drawing 26 is the conventional parallel of the cipher-processing circuit in the CBC mode. This figure (a) An enciphering circuit is shown. When it can set to drawing 24 in this figure, it is M1 the first 8 bits of a plaintext similarly. If inputted, an exclusive OR with the initial vector IV stored in the register 75 will be taken by the EXOR circuit 76, and the result will be inputted into the f function 77.

[0017] [ in order to perform all of 16 steps of conversion using f function as an f function 77, may have 16 steps of conversion circuits but ] in order to reduce the amount of hardware, for example 16 steps of conversion is also realizable by having four steps of conversion circuits and carrying out the loop of the intermediate result of processing 4 times through a register 75 and the EXOR circuit 76. In this case, the contents of storing of a register 75 can be outputted to the f function 77 as it is between processings of that loop by setting 64 bits of values of the input to the EXOR circuit 76 to "0" as opposed to all.

[0018] In addition, [ since it was easy here, only the f function 77 was shown, but ] The initial transposition IP included in the encryption processing E52 of drawing 24 can be considered that it shall be contained in processing of f function of the 1st step, and exchange of 32 bits of the last right and left and reverse transposition IP-1 are contained in the 16th step of processing of f function.

[0019] And 64 bits which it is at the end time of the 4th loop, is outputted from the f function 77, and is stored in a register 75 are the block C1 of the beginning of a cipher. While being outputted by carrying out, it is the block M2 next to an input plaintext. It is M2 at the input time.

It is used in order to take an exclusive OR. Since the following processings are the same as the explanation to drawing 24 , the explanation is omitted.

[0020] Drawing 26 (b) It is the conventional parallel of the CBC decoder circuit. In this figure, it is the block C1 of the beginning of a cipher. An input will give the input to the f function 81 through a selector 80.

[0021] This f function 81 is for performing decoding processing D53 in drawing 24 , and is (a). if it shall have four steps of conversion circuits similarly It is the input cipher block C1 after the end of four operation through a register 82 and a selector 80 of a loop. A decoded result will be outputted to a register 82, and conversion of 32 bits of right and left and reverse transposition IP-1 shall be included in the initial transposition IP and the 16th step of conversion at the 1st-step conversion.

[0022] Block C1 of the beginning of the cipher stored in the register 82 It is inputted into the EXOR circuit 84, an exclusive OR with the initial vector IV stored in the register 83 is taken, and a decoded result is the block M1 of the beginning of a plaintext. It is outputted by carrying out. At this time, it is the next block C2. In order to take a decoded result and an exclusive OR, it is the block C1 of the beginning of an input cipher. It is stored in a register 83. Block C2 next to a cipher Since decoding processing to an input is similarly performed in drawing 24 , the explanation is omitted below.

[0023] Drawing 27 is the conventional parallel of the cipher-processing circuit in CFB mode. This figure (a) An enciphering circuit and (b) A decoder circuit is shown. Drawing 27 (a) It sets and is M1 the first 8 bits of a plaintext. Corresponding to an input, the value of the initial vector IV first stored in the register 86 is outputted to the f function 87.

[0024] The f function 87 performs the encryption processing E67 E52 in drawing 25 , i.e., the encryption processing in drawing 24 , and same processing, and it has four steps of conversion circuits similarly in drawing 26 . 16 steps of conversion by four processings through a selector 90 and a register 86 of a loop ([ and ] in drawing 26 similarly the first stage) 8 bits of lefts are outputted to the EXOR circuit 91 among 64 bits as a processing result including exchange of 32 bits of right and left, and reverse transposition -- 8 bit M1 of an input plaintext an exclusive OR -- it can take -- block C1 of the beginning of a cipher \*\*\*\*\* -- it is outputted.

[0025] As mentioned above, the initial vector IV stored in the register 86 is M1. Although outputted to the f function 87 corresponding to an input, at this time, it is shifted to the 8-bit left, namely, left 8 bits are thrown away, and those contents are stored in a register 88. The number of bits is 56 bits. Only 8 bits of most left-hand side of the output after the 4th loop of the f function 87 are inputted into the EXOR circuit 89. Input M1 An exclusive OR is taken and 8 bits of the result are stored most in right-hand side as 64-bit data through a selector 90 at a register 86 in form that 56 bits stored in the register 88 are arranged on left-hand side. These 64 bits will be outputted to the encryption processing E67 of drawing 25 corresponding to the

input of the next block M2 of a plaintext. M2 Since operation to the following blocks is the same, the following explanation is omitted.

[0026] Drawing 27 (b) It sets and is C1 the first 8 bits of a cipher. at the input time The initial vector IV stored in the register 93 is given to the f function 94. Conversion using 16 steps of f functions etc. is performed by four processings of a loop which mind a selector 95 and a register 93 like the above-mentioned. 8 bits of lefts are outputted to the EXOR circuit 97 among 64 bits as an output of the encryption processing E71 of drawing 25 , and it is the input cipher block C1. An exclusive OR is taken and it is a plaintext M1. It is outputted.

[0027] When the value of the initial vector IV as contents of the register 93 is outputted to the f function 94, simultaneously, it acts as a 8-bit left shift of the contents, and 56 bits is stored in a register 96. And it is at the end time of the 4th loop, and, in left-hand side and 8 bits of input ciphers, the 56 bits are C1. In the form arranged on right-hand side, 64-bit data is stored in a register 93 through a selector 95. This data is given to the f function 94 corresponding to the input of the next cipher block C2. Cipher block C2 Since it is the same about the decoding processing to the following, the explanation is omitted.

[0028]

[Problem to be solved by the invention] In order to accelerate such DES encryption processing, it aims at the improvement in the speed of 16 steps of transform processing which used f function especially conventionally. the conversion circuit of f function -- as much as possible -- a number of stages -- [ it connects mostly, and / loop loss is reduced or ] The procedure of the conversion containing f function was changed and the method of reducing the delay to 1 time of a loop, or having put in, for example in the CBC encryption processing into the processing cycle of encryption of the mode processing corresponding to the operation mode, i.e., a loop, and reducing the number of clocks required for processing was used. It is indicated by the following United States patent about these methods.

[0029] U.S. the conversion using Patent 5, 381, and 480, Butter et al."System for Translating Encrypted Data", however f function -- as much as possible -- a number of stages -- though it connects mostly The more it increased the number of stages, the more the amount of hardware increased, and since cost became large, there was a problem that a number of stages could also seldom be increased.

[0030] Next, as the operation mode of a DES block cipher, the four operation modes, ECB, CBC, CFB, and OFB, exist as mentioned above. ECB mode only combines DES encryption processing and decoding processing, and the special mode processing circuit for the operation mode is unnecessary. In OFB mode, within the processing loop of DES encryption Processing to the next block can be performed as it is using the processing result of a pre- block, and it is not necessary to take [ as special processing corresponding to a mode ] generating of delay into consideration to a processing loop that what is necessary is just to take an exclusive OR

between the final disposal result of DES, and input block data.

[0031] On the other hand, it is necessary to take the exclusive OR of the value of the initial vector IV, and input block, for example, and to perform DES processing to this value in the CBC mode. Moreover, it is necessary to perform the left shift of the value of an initial vector at the time of the renewal of contents of the register which stored the initial vector IV in encryption processing, and to set the value of an exclusive OR with a part of processing result of DES processing, and input block to right-hand side in CFB mode.

[0032] That is, the mode processing to CBC is the beginning of DES processing, and there is a difference that mode processing to CFB is performed at the last. These the processings of both are included in the loop of DES processing.

[0033] Although it was desirable to constitute the circuit which can perform all these four operation modes as a processing circuit which processes DES encryption, there was a problem that it was difficult to constitute the processing circuit especially used common to the CBC mode and CFB mode in the former. This originates in the mode processing corresponding to these two modes being [ one side / another side ] contained at the end at the beginning of the processing loop of DES processing.

[0034] Especially this invention aims at offering the cipher-processing circuit which can perform both the CBC mode and CFB mode, and reducing the processing delay in the circuit as much as possible.

[0035]

[Means for solving problem] Drawing 1 is a principle configuration block figure corresponding to the 1st embodiment of this invention. This figure is a configuration block figure of a high-speed cipher-processing circuit which can perform DES encryption processing in both the CBC mode and CFB mode, for example as the operation mode and which processes block encryption.

[0036] In drawing 1 , the data storage means 1 stores the initial vector used for block encryption processing, a processing mean value, or the processing final result, and is a register.

[0037] The function processing means 2 performs two or more steps of conversion using f function as function processing used within block encryption processing, for example, DES processing. It has the mode processing means 3 between the data storage means 1 and the function processing means 2, and it performs processing corresponding to the operation mode, for example, the CBC mode, or CFB mode of block encryption using the data stored in the data storage means 1.

[0038] In processing in CFB mode, 16 steps of conversion using the nonlinear function as the above-mentioned function processing in the input-block data encryption to a DES code is performed in a step, for example, 4 times, by the function processing means 2. In that case, in

one function processing, four steps of conversion using a nonlinear function is performed.

[0039] The execution result at each time except the last 1 time of processing of these multiple times is stored in the data storage means 1 by the function processing means 2. And since the encryption block data to input block data is outputted using 1 time of this last processing result, it has an encryption data output means further.

[0040] Moreover, while mode processing is performed and that execution result is outputted to the function processing means 2 by the mode processing means 3 before execution of the time of the beginning of processing of these multiple times, it has further a mode processing result storing means by which a part of execution result of mode processing is stored. And while a part of execution result of the above-mentioned mode processing is outputted to the data storage means 1 by the mode processing result storing means at the time of execution of the time of the last of processing of the multiple times by the function processing means 2, a part of execution result is outputted to the data storage means 1 by the function processing means 2.

[0041] In the time of execution of encryption processing in the CBC mode, in CFB mode, similarly, 16 steps of conversion using the nonlinear function as function processing in encryption of the block data to the above-mentioned DES code divides into multiple times, and is performed. And the execution result of each time of processing of these multiple times is stored in the data storage means 1 by the function processing means 2.

[0042] Moreover, before execution of the time of the beginning of processing of the multiple times by the function processing means 2, mode processing using the block data inputted by the mode processing means 3 and the data stored in the data storage means is performed, and the result of the mode processing is outputted to the function processing means 2.

[0043] Drawing 2 is a principle configuration block figure corresponding to the 2nd embodiment of this invention. In addition to the 1st mode processing means 4 corresponding to the mode processing means 3 of drawing 1, in this figure, it has between the function processing means 2 and the data storage means 1, and the 2nd mode processing means 5 which performs 2nd processing corresponding to the operation mode of block encryption is added.

[0044] In the 2nd embodiment of this invention although the cipher-processing circuit can perform the operation mode in both CFB mode and the CBC mode similarly in the 1st embodiment Fundamental mode processing in CFB mode is performed by the 2nd mode processing means 5 to the 1st mode processing means 4 performing the fundamental portion of the mode processing in the CBC mode.

[0045] [ conversion ] in CFB mode although it divides into multiple times similarly and 16 steps of conversion using the nonlinear function by the function processing means 2 is performed also in the 1st embodiment The data stored in the data storage means 1 in the first time is given to the function processing means 2 as it is, without performing special processing by the

1st mode processing means 3. The processing result at each time except the time of the last of processing of the multiple times by the function processing means 2 is stored in the data storage means 1.

[0046] While the encryption block data to input block data is similarly outputted in the 1st embodiment using the processing result of the last round by the function processing means 2. By the 2nd mode processing means, it is performed by the 2nd mode processing using the part and input data block of the processing result, and The result, Before execution of the time of the beginning of multiple times, some data stored in the data storage means 1 is stored in the data storage means 1, and it is used for encryption processing of the following block data.

[0047] In the CBC mode, according to the mode processing means 3 against the 1st embodiment, same processing is performed by the 1st mode processing means 4, and also in the 1st embodiment, same processing is performed as a result, without the 2nd mode processing means performing any operation substantially.

[0048] While being at the end time of encryption processing of one block data and outputting a processing result in the cipher-processing method of processing DES block encryption, for example, in the cipher-processing method of this invention The processing mean value in the encryption processing or the processing final result is stored in the register in which the initial vector was stored.

[0049] And the operation corresponding to the operation mode of the block encryption is performed between the data stored in the register, its one block data, or the following block data following the one block. Furthermore, the result of an operation is used as a value equivalent to the initial vector in the encryption processing to the following block data following the one block, and encryption processing to the following block data is performed.

[0050] In the encryption processing method of this invention, the initial vector is first stored in the above-mentioned register. After the operation corresponding to the operation mode is first performed between input block data in the CBC mode as opposed to the initial vector In CFB mode, without performing substantially, corresponding to the result of an operation, 16 steps of conversion using a nonlinear function divides the special operation corresponding to the operation mode into multiple times, and it is performed. The method of this encryption processing is used in the 1st above-mentioned embodiment.

[0051] As explained above, according to this invention, the cipher-processing circuit which performs both the CBC mode and CFB mode is realizable. Moreover, it becomes possible to communalize the mode processing means corresponding to two modes in the 1st embodiment.

[0052]

[Mode for carrying out the invention] Drawing 3 shows the composition of the enciphering circuit in the CFB mode used in the 1st embodiment of this invention. This figure is drawing 27

(a). It is drawing 27 (a) in order to use it combining the enciphering circuit in the CBC mode, although it corresponds to the shown conventional parallel. The EXOR circuit 89 set and used is omitted, instead the EXOR circuit 12 is used for the output side of the register 11.

[0053] The initial vector IV is stored in the register 11 in drawing 3. 8 bit M1 of the beginning of a plaintext At the input time, the value of the initial vector IV is given to the f function (processing circuit) 13 from a register 11. At this time, the value of the initial vector IV stored in the register 11 is given to the f function 13 from a register 14 as it is by outputting "0", for example to the EXOR circuit 12.

[0054] The f function 13 is drawing 27 (a). If it can set, it will consist of a conversion circuit using four steps of f functions similarly, and 16 steps of conversion using f function shall be performed by four processings through a selector 15 and a register 11 of a loop. Moreover, the initial transposition IP, exchange of 32 bits of right and left to the result of the 16th step, and reverse transposition IP-1 shall be included as mentioned above.

[0055] 8 bits of lefts are given to the EXOR circuit 18 among the outputs of the f function 13 after 4 times of loops, and it is the input plaintext block M1. An exclusive OR is taken and it is the first cipher block C1. It is outputted by carrying out.

[0056] Block M1 [ 56 bits of left-hand side ] among the contents although 8 bits of right-hand side is outputted to the register 17 side from a register 11 as it is in the form given through the EXOR circuit 12 when the initial vector IV is outputted from a register 11 corresponding to an input The 64 bits are stored in a register 17, after 8 bit shifts are carried out and 8 bits of most left-hand side are thrown away. And 64-bit data is stored in a register 11 through a selector 15 in form that are at the end time of 4 times of loops, and 56 bits of the contents are stored in left-hand side, and 8 bits of lefts of the outputs of the f function 13 are stored in right-hand side. Moreover, input-block M1 8 bit is stored in a register 14 at this time. 64 bits stored in the register 11 here are the following input block M2, if it remains as it is. It cannot use as an object of DES encryption processing to the f function 13 at the input time.

[0057] Block M2 At the input time, 8 bits of right-hand side is outputted to the EXOR circuit 12 among the contents of storing of a register 11. An exclusive OR with 8 bits of the first block M1 stored in the register 14 is taken, and it is inputted into the f function 13 with 56 bits of left-hand side outputted as it is from a register 11. 8 bit shifts are carried out to the left like the above-mentioned, and 56 bits of 64 bits [ which becomes an input to the f function 13 substantially at this time ] are stored in a register 17. This is the next block M3. It is for the data origination to the f function 13 at the input time.

[0058] It is at the end time of 4 times of the loops through the f function 13, a selector 15, and a register 11, the exclusive OR of 8 bits of lefts and input M2 8 bit is taken by the EXOR circuit 18 among the outputs of the f function 13, and it is the block C2 next to a cipher sequence. It is outputted by carrying out. The contents of storing of a register 14 shall not be outputted

between this four loop processing, but "0" shall be given to the EXOR circuit 12. Since the same processing as the following is performed, the explanation is omitted.

[0059] Drawing 4 is the configuration block figure of the cipher-processing circuit as the 1st embodiment of this invention. This figure is a configuration block figure of the cipher-processing circuit which performs processing in the mode in both CFB mode and the CBC mode. CFB mode encryption operation using this circuit is explained using drawing 5 - drawing 9 .

[0060] Drawing 5 is the explanatory view of CFB mode encryption operation (the 1). The dashed line has shown the portion which is a solid line and is not used about the connection portion of the circuit used here. It is the first plaintext block M1 first. The initial vector IV stored in the register 21 corresponding to the input (from \*\*) gives the f function 26 through selectors 22 and 23 and the EXOR circuits 24 and 25, and it is \*\*\*\*. At this time, selectors 22 and 23 choose the output from a register 21, and the data (56 bits and 8 bits) of "0" is inputted through selectors 31 and 32 to the EXOR circuits 24 and 25. For this reason, the value of the initial vector IV stored in the register 21 is given to the f function 26 as it is. Moreover, 8 bit shifts of the value of IV stored in the register 21 are only carried out, and it is substantially stored in a register 28 as it is. And the processing result of four steps of conversion circuits contained in the f function 26 is stored in a register 21 through a selector 27.

[0061] Drawing 6 is the explanatory view of CFB mode encryption operation (the 2). This figure is an explanatory view of processing operation in the 2nd time and the 3rd loop among 4 times of the loops through the f function 26, a selector 27, and a register 21. Since selectors 22 and 23 choose the output from a register 21 in these loops and selectors 31 and 32 are outputting "0" and "0" to the EXOR circuits 24 and 25, respectively, [ 56 bits ] [ 8-bit ] The data of a register 21 is given to the f function 26 as it is, and four steps of processing results are stored in a register 21 through a selector 27.

[0062] Drawing 7 is the explanatory view of CFB mode encryption operation (the 3). In this figure, the processing result of the 3rd loop stored in the register 21 is given to the f function 26 as it is through selectors 22 and 23 and the EXOR circuits 24 and 25 like the above-mentioned. Since the output of the f function 26 is the encryption processing result of the initial vector IV which used 16 steps of f functions, Block M1 of the beginning of the plaintext which 8 bits of the left is taken out and given to input \*\* An OR with 8 bits is taken by the EXOR circuit 30, and it is the block C1 of the beginning of a cipher. It carries out and is outputted from output \*\*. Moreover, plaintext block M1 8 bits is stored in a register 33.

[0063] 64-bit data is stored in a register 21 through a selector 27 in form that 56 bits simultaneously stored in the register 28 are stored in left-hand side, and 8 bits of lefts are stored in right-hand side among the outputs of the f function 26.

[0064] Drawing 8 is the explanatory view of CFB mode encryption operation (the 4). Plaintext



block M2 of an input [ here ] The first processing in which it corresponds is performed. The 64-bit data first stored in the register 21 by (the 3) is given to two EXOR circuits 24 and 25 through selectors 22 and 23. At this time, a selector 31 is the block M1 of the beginning of the data with which 32 is stored in the register 33 in "0", i.e., an input plaintext. although data is outputted and the EXOR circuit 24 outputs 56 bits of left-hand side to the f function 26 as it is among 64 bits [ 56-bit ] A circuit 25 will output 8 bits of right-hand side, and an M18 bit exclusive OR, and 64 bits of the result are block M2. It is given to the f function 26 at the beginning of four processings of a loop corresponding to an input. Moreover, substantially, 8 bit shifts of this input are carried out, and it is stored in a register 28. And the result of four steps of transform processing (initial transposition is included) in the f function 26 is stored in a register 21 through a selector 27.

[0065] Drawing 9 is the timing chart of CFB mode encryption operation. encryption \*\*\*\* which explained the number over a flow of operation by drawing 5 - drawing 8 -- the 1- it is a number corresponding to the 4. Block M1 of the beginning of a plaintext To an input, corresponding to four clocks, processing of the 1, its 2, and the 2 and its 3 is performed, and it is the block C1 of the beginning of an output. It is obtained. The following input block M2 To the following, corresponding to four clocks, processing of the 4, its 2, and the 2 and its 3 is performed, and it is the output block C2. The following is obtained. In addition, cipher block C1 As drawing 7 explained, they are the output of the f function 26, and the plaintext block M1 to the last of the 4th loop. Although a short-time output is carried out as a result of an exclusive OR, the output period is shown in the figure by about 1 clock. If a latch circuit is prepared before output terminal \*\*, it is also possible to extend this period before the next cipher block output.

[0066] Drawing 10 is the explanatory view of CFB mode decode operation (the 1). In this figure, it is the block C1 of the beginning of a cipher. The value of the initial vector IV stored in the register 21 corresponding to the input is given to the f function 26 as it is through selectors 22 and 23 and the EXOR circuits 24 and 25. Moreover, the initial vector IV is stored in a register 28 only by acting as a 8-bit left shift. And the result is stored in a register 21 through a selector 27 after the end of four steps of processings by f function including initial transposition.

[0067] Drawing 11 is the explanatory view of CFB mode decode operation (the 2). This figure is an explanatory view of processing operation of the 2nd loop of 4 times of the loops as processing which use f function, and the 3rd loop. The 64-bit data stored in the register 21 is given to the f function 26 through selectors 22 and 23 and the EXOR circuits 24 and 25, and the output of the f function 26 is stored in a register 21 through a selector 27.

[0068] Drawing 12 is the explanatory view of CFB mode decode operation (the 3). This figure shows processing operation of the 4th loop which uses f function. The 64-bit data stored in the register 21 is given to the f function 26 through selectors 22 and 23 and the EXOR circuits 24

and 25. Four steps of conversion and the exchange of 32 bits of right and left, and reverse transposition IP-1 after processing [ using f function ] The first cipher block C1 which 8 bits of lefts are given to the EXOR circuit 30 among 64 bits as the processing result, and is inputted from input \*\*. An exclusive OR with 8 bits is taken and the result is the block M1 of the beginning of a plaintext from output \*\*. It is outputted by carrying out. Moreover, to left-hand side, 56 bits simultaneously stored in the register 28 are the input cipher block C1. In the form arranged on right-hand side, 8 bits is stored in a register 21 as 64-bit data through a selector 27.

[0069] Drawing 13 is the timing chart of CFB mode decode operation. the number in a flow of operation -- the 1- as drawing 10 - drawing 12 -- the processing of 3 is shown. Block C1 of the beginning of an input cipher To an input, in four clocks, processing of the 1, its 2, and the 2 and its 3 is performed, and it is the block M1 of the beginning of an output plaintext. It is outputted. The next input cipher block C2 Same operation is completely performed also to the following inputs.

[0070] Drawing 14 is the explanatory view of the CBC mode encryption operation (the 1). In the CBC mode, an input plaintext block is inputted from input \*\*, and an output cipher block is outputted from output \*\*. Before long, 56 bits of lefts are given to a selector 31, and, as for the plaintext block M1 64 bit of the beginning first inputted from input \*\*, 8 bits of rights are given to 32. And as for a selector 31, the 56 bits are given to the EXOR circuit 24, and 32 gives 8 bits to 25.

[0071] The EXOR circuits 24 and 25 are given through selectors 22 and 23, and the initial vector IV stored in the register 21 on the other hand is the plaintext input block M1 of the initial vector IV and the beginning. An exclusive OR is taken and the result is given to the f function 26. And the result of the conversion using four steps of f functions including the initial transposition IP is stored in a register 21 through a selector 27.

[0072] Drawing 15 is the explanatory view of the CBC mode encryption operation (the 2). This figure is an explanatory view of operation in the 2nd time and the 3rd loop among 4 times of the loops which use f function. The 64-bit data stored in the register 21 is given to two EXOR circuits 24 and 25 through selectors 22 and 23. Since selectors 31 and 32 are outputting "0" and "0", respectively at this time, that 64-bit data is given to the f function 26 as it is. [ 56 bits ] [ 8-bit ] And the result of four steps of transform processing using f function is stored in a register 21 through a selector 27.

[0073] Drawing 16 is the explanatory view of the CBC mode encryption operation (the 3). This figure illustrates operation of the 4th loop which uses f function. The 64-bit data stored in the register 21 is given to the f function 26 as it is through selectors 22 and 23 and the EXOR circuits 24 and 25 like the above-mentioned. Four steps of conversion and the exchange of 32 bits of right and left using f function, and the processing result of reverse transposition IP-1 are

stored in a register 21 through a selector 27. the 64-bit data stored in the register 21 minds the EXOR circuit 34 -- as it is -- block C1 of the beginning of output \*\* to an output cipher \*\*\*\*\* -- \*\*\*\* -- last \*\* At this time, the selector 35 is outputting "0" to the EXOR circuit 34. [ 64-bit ] Moreover, block C1 of the beginning of the output cipher stored in the register 21 The next plaintext block M2 It is the input block M2 by encryption operation (the 1) at the input time. An exclusive OR will be taken like the above-mentioned.

[0074] Drawing 17 is the timing chart of the CBC mode encryption operation. Input plaintext block M1 Corresponding to an input, in four clocks, operation of the 1, its 2, and the 2 and its 3 is performed, and it is the block C1 of the beginning of an output cipher. It is outputted.

Plaintext block M2 Operation to the following inputs is completely the same.

[0075] Drawing 18 is the explanatory view of the CBC mode decode operation (the 1). The initial vector IV is stored in the register 37 in the CBC mode decode operation. The first cipher block C1 If inputted from input \*\*, at a selector 31, by 56 bits of the left-hand side, and 32, 8 bits of right-hand side will be chosen, and two EXOR circuits 24 and 25 will be given. At this time, both the selectors 22 and 23 are outputting "0" and "0", and, for this reason, are the input cipher blocks C1. 64 bits is given to the f function 26 as they are. [ 56 bits ] [ 8-bit ] And four steps of processing results using f function including the initial transposition IP are stored in a register 21 through a selector 27.

[0076] Drawing 19 is the explanatory view of the CBC mode decode operation (the 2). This figure is an explanatory view of operation in the 2nd time and the 3rd loop among 4 times of the loops which use f function. The 64-bit data stored in the register 21 in this figure is given to the f function 26 as it is through selectors 22 and 23 and the EXOR circuits 24 and 25, and the processing result of four steps of conversion using f function is stored in a register 21 through a selector 27. Moreover, the initial vector IV stored in the register 37 is stored in a register 36 at the time of processing of the 3rd loop (the 2nd time is sufficient).

[0077] Drawing 20 is the explanatory view of the CBC mode decode operation (the 3). The 64-bit data stored in the register 21 as 4th loop in this figure is given to the f function 26 as it is, and four steps of conversion and the exchange of 32 bits of right and left using f function, and the processing result of reverse transposition IP-1 are given to a register 21 through a selector 27. The EXOR circuit 34 is given, an exclusive OR with the initial vector IV stored in the register 36 is taken, and the 64-bit data stored in the register 21 is the block M1 of the beginning of an output plaintext from output \*\*. It is outputted by carrying out. Moreover, it is the next cipher block C2 at this time. As data for outputting to the EXOR circuit 34 in operation receiving (that 3), it is the block C1 of the beginning of an input cipher. It is stored in a register 37.

[0078] Drawing 21 is the timing chart of the CBC mode decode operation. In this figure, it is the block C1 of the beginning of a cipher. To an input, in four clocks, operation of the 1, its 2, and

the 2 and its 3 is performed, and it is the block M1 of the beginning of a plaintext. It is outputted. Operation to the input of next block C2 ... of a cipher is completely the same below. [0079] Drawing 22 is the configuration block figure of the cipher-processing circuit which performs two operation, the CFB mode as the 2nd embodiment of this invention, and the CBC mode. [ in order that two EXOR circuits 24 and 25 might take the exclusive OR to 8 bits of left-hand side also in encryption operation in not only the CBC mode but CFB mode in the 1st embodiment of drawing 4 among the 64-bit data which should be given to the f function 26, were used, but ] In drawing 22 , there is a difference with this exclusive OR fundamental to the point taken by the EXOR circuit 45. For this reason, in the EXOR circuits 24 and 25, as for the selectors 22 and 23 of drawing 4 , one selector 41 is realized, and one EXOR circuit 42 and selectors 31 and 32 are realized by one selector 43.

[0080] Since CFB mode encryption operation has fundamentally the difference with the 2nd embodiment of drawing 22 , and the 1st embodiment of drawing 4 , it explains drawing 22 focusing on the operation. In drawing 22 , the initial vector IV stored in the register 21 corresponding to the above-mentioned CFB mode encryption operation (the 1) is given to the f function 26 as it is through a selector 41 and the EXOR circuit 42, and the processing result is stored in a register 21 through a selector 46. Operation in encryption operation (the 2) is also the same.

[0081] The first plaintext block M1 into which it is equivalent to encryption operation (the 3), and 8 bits of lefts are inputted from input \*\* among the outputs of the f function 26 as 4th loop. An exclusive OR is taken by the EXOR circuit 30 and it is outputted as the first cipher block C1 from output \*\*. [ 56 bits simultaneously stored in the register 44 / left-hand side / moreover, form that the result by which the exclusive OR of 8 bits of lefts and input-block M1 8 bit was taken by the EXOR circuit 45 is arranged at right-hand side ] among the outputs of the f function 26 64-bit data is stored in a register 21 through a selector 46. An exclusive OR with an input is already taken among the data stored in this register 21, and 8 bits of rights are the next plaintext blocks M2. Corresponding to an input, it will be given to the f function 26 as it is through a selector 41 and the EXOR circuit 42.

[0082] Also in the 1st embodiment, since it is substantially the same, operation of CFB mode decode, the CBC mode encryption, and the CBC mode decode omits the explanation. Drawing 23 is the explanatory view of the processing included in one clock in the 1st embodiment and 2nd embodiment. In each of the 2nd embodiment shown in the 1st embodiment and drawing 22 which were shown in drawing 4 The point which 16 steps of conversion using the f function 26 is performed by dividing into every four-step four clocks, and needs four clocks for the encryption processing to one block data is the same also in which embodiment. However, in the 1st embodiment, it becomes short, it becomes possible to gather the speed of a clock as a result, and encryption processing can be accelerated rather than the sum total of the time

delay of the processing performed within one clock can set to the 2nd embodiment.

[0083] In the 1st loop including the 1-4th step of conversion by f function of drawing 23 Delay according to the output from a register 21 at drawing 4 corresponding to the 1st embodiment of the lower berth, In addition to the delay of conversion (initial transposition is included) using delay by selectors 22 and 23, delay of processing by the EXOR circuits 24 and 25, and four steps of f functions, and delay by a selector 27, delay by the setup of the data in a register 21 will be included in one clock.

[0084] In the circuit of drawing 22 corresponding to the 2nd embodiment of an upper case to it Delay by the EXOR circuit 45, delay by a selector 27, and delay by the data setup of a register 21 are included after delay of the data output by a register 21, delay by a selector 41, delay by the EXOR circuit 42, and delay by four steps of conversion using f function. That is, compared with the 1st embodiment, delay by the EXOR circuit 45 is included too many, in the 2nd embodiment, compared with the 1st embodiment, only the part becomes long and the cycle of a clock will be inferior to the 1st embodiment in it in respect of improvement in the speed. However, if this point is removed, the cipher-processing circuit which performs the operation mode in both CFB mode and the CBC mode also in the 2nd embodiment will be realized.

[0085]

[Effect of the Invention] As explained to details above, according to this invention, it becomes possible to realize the cipher-processing circuit which can perform the operation mode in both the CBC mode of a DES code, and CFB mode. Moreover, by moving the mode processing in CFB mode to the beginning of processing to the following block data in the 1st embodiment It becomes possible to use the portion which performs processing in the CBC mode, and a common portion, and the maximum delay of a processing loop can be shortened and high speed processing becomes possible. Moreover, the place which the reduction of the circuits of an exclusive OR of is also attained simultaneously, can decrease in number the amount of hardware, and contributes to the improvement in practicality of a cipher-processing method is large.

---

#### [Brief Description of the Drawings]

[Drawing 1] It is a principle configuration block figure corresponding to the 1st embodiment.

[Drawing 2] It is a principle configuration block figure corresponding to the 2nd embodiment.

[Drawing 3] It is the block diagram showing the composition of the CFB mode enciphering circuit used in the 1st embodiment of this invention.

[Drawing 4] It is the block diagram showing the composition of the cipher-processing circuit as the 1st embodiment of this invention.

[Drawing 5] It is the explanatory view of CFB mode encryption operation (the 1) in the 1st embodiment.

[Drawing 6] It is the explanatory view of CFB mode encryption operation (the 2) in the 1st embodiment.

[Drawing 7] It is the explanatory view of CFB mode encryption operation (the 3) in the 1st embodiment.

[Drawing 8] It is the explanatory view of CFB mode encryption operation (the 4) in the 1st embodiment.

[Drawing 9] It is the timing chart of CFB mode encryption operation.

[Drawing 10] It is the explanatory view of CFB mode decode operation (the 1) in the 1st embodiment.

[Drawing 11] It is the explanatory view of CFB mode decode operation (the 2) in the 1st embodiment.

[Drawing 12] It is the explanatory view of CFB mode decode operation (the 3) in the 1st embodiment.

[Drawing 13] It is the timing chart of CFB mode decode operation.

[Drawing 14] It is the explanatory view of the CBC mode encryption operation (the 1) in the 1st embodiment.

[Drawing 15] It is the explanatory view of the CBC mode encryption operation (the 2) in the 1st embodiment.

[Drawing 16] It is the explanatory view of the CBC mode encryption operation (the 3) in the 1st embodiment.

[Drawing 17] It is the timing chart of the CBC mode encryption operation.

[Drawing 18] It is the explanatory view of the CBC mode decode operation (the 1) in the 1st embodiment.

[Drawing 19] It is the explanatory view of the CBC mode decode operation (the 2) in the 1st embodiment.

[Drawing 20] It is the explanatory view of the CBC mode decode operation (the 3) in the 1st embodiment.

[Drawing 21] It is the timing chart of the CBC mode decode operation.

[Drawing 22] It is the block diagram showing the composition of the cipher-processing circuit as the 2nd embodiment of this invention.

[Drawing 23] It is the figure showing comparison of the processing performed within one clock in the 1st embodiment and 2nd embodiment.

[Drawing 24] It is the basic explanatory view of the encryption processing in the CBC mode.

[Drawing 25] It is the basic explanatory view of encryption processing in CFB mode.

[Drawing 26] It is the figure showing the conventional parallel of the cipher-processing circuit in

the CBC mode.

[Drawing 27] It is the figure showing the conventional parallel of the cipher-processing circuit in CFB mode.

[Explanations of letters or numerals]

1 Data Storage Means

2 Function Processing Means

3 Mode Processing Means

4 1st Mode Processing Means

5 2nd Mode Processing Means

21, 28, 33, 36, 37, 44 Register

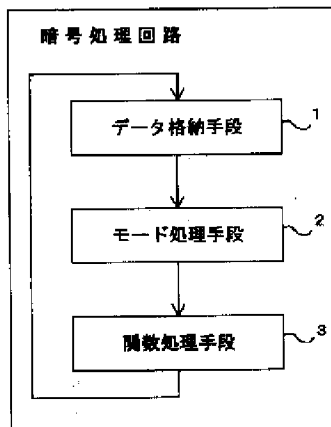
22, 23, 27, 31, 32, 35, 41, 46 Selector

24, 25, 30, 34, 42, 45 EXOR circuit

26 F Function (Processing Circuit)

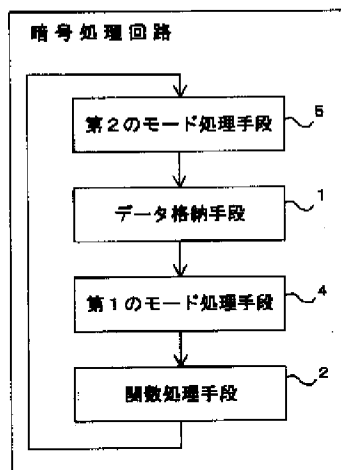
[Drawing 1]

第1の実施形態に対応する  
原理構成ブロック図



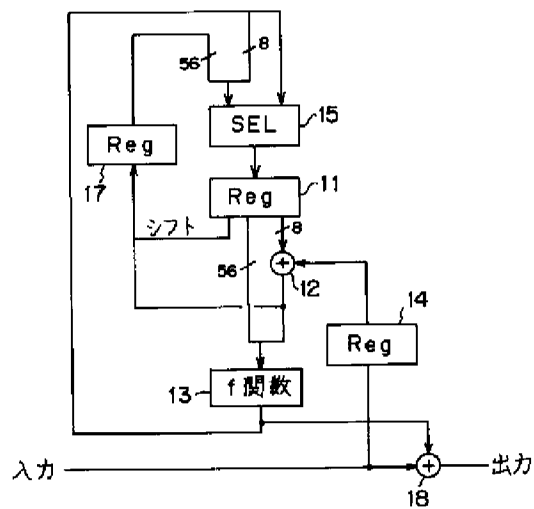
[Drawing 2]

第2の実施形態に対応する  
原理構成ブロック図



[Drawing 3]

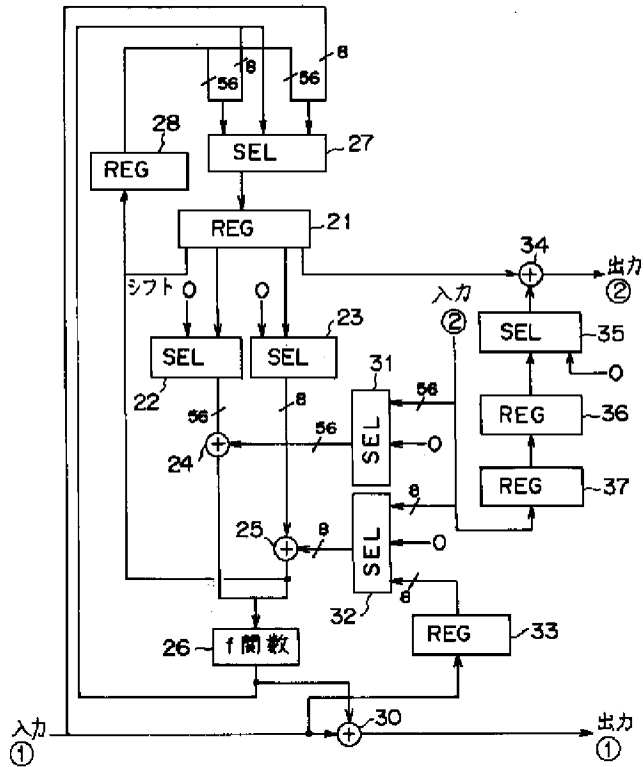
本発明の第１の実施形態において使用される  
CFBモード暗号化回路の構成を示すブロック図



[Drawing 4]



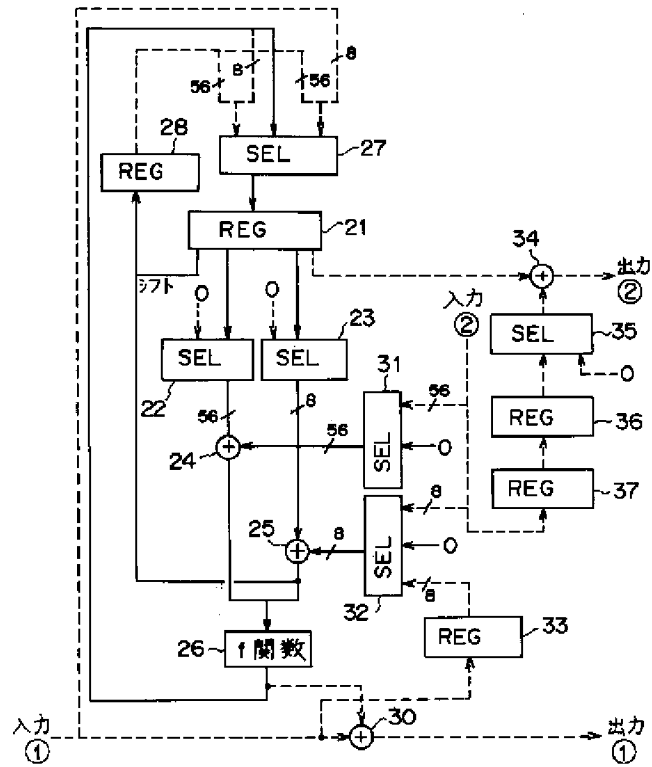
本発明の第1の実施形態としての  
暗号処理回路の構成を示すブロック図



[Drawing 5]

第1の実施形態におけるCFBモード暗号化動作

(その1) の説明図

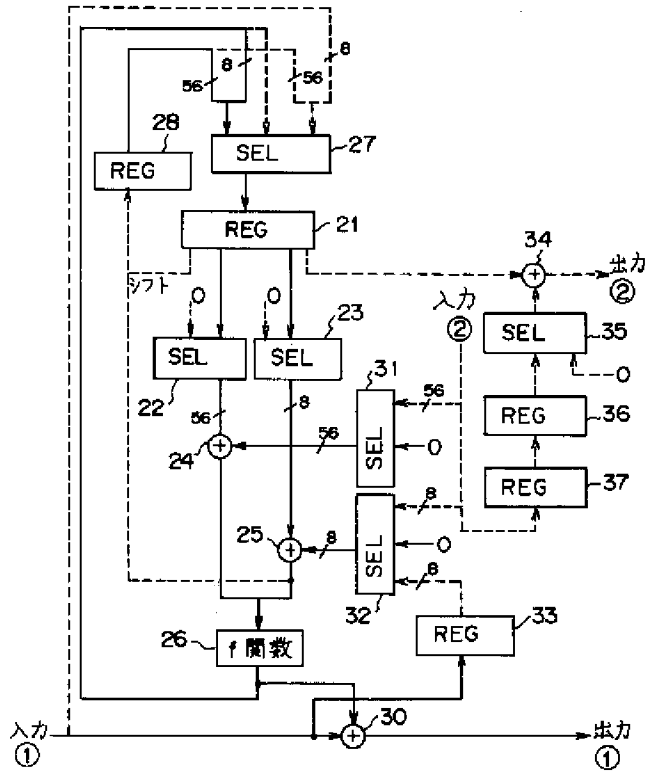


[Drawing 6]

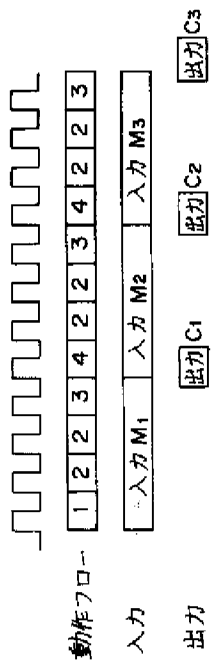


## 第1の実施形態におけるCFBモード暗号化動作

(その3)の説明図

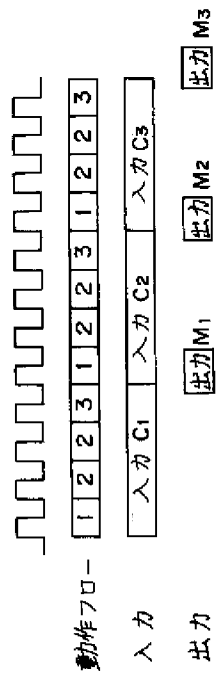


[Drawing 9]

CFBモード暗号化動作の  
タイミングチャート

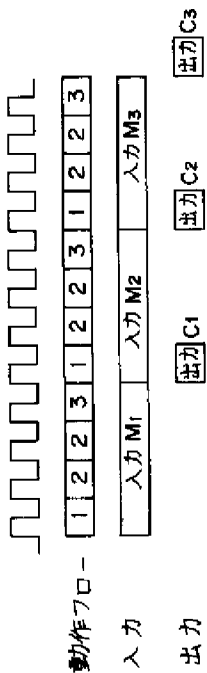
## [Drawing 13]

CFBモード復号動作の  
タイミングチャート



## [Drawing 17]

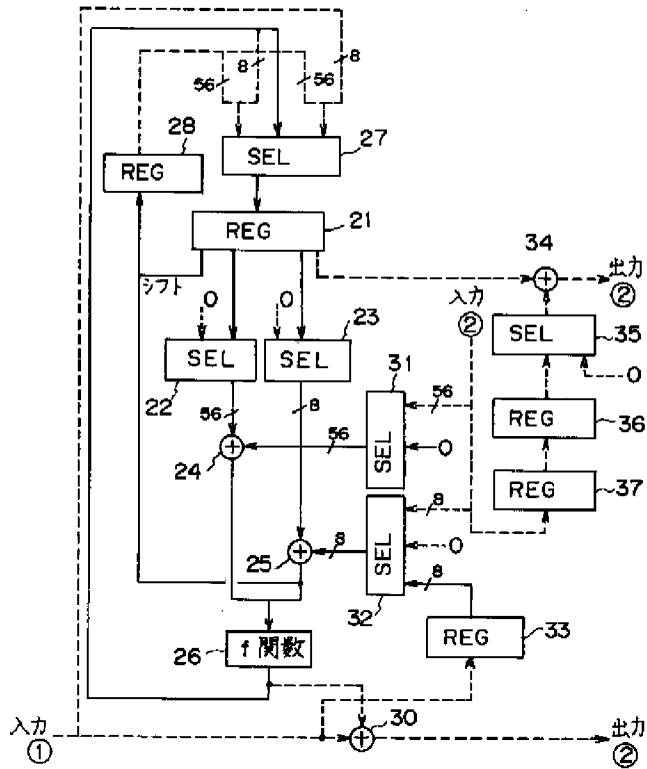
CBCモード暗号化動作の  
タイミングチャート



## [Drawing 8]

第1の実施形態におけるCFBモード暗号化動作

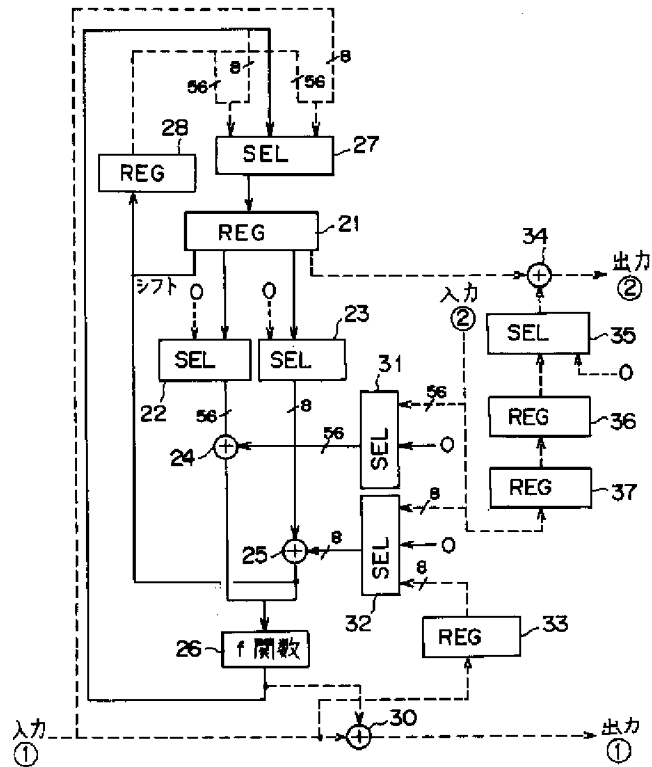
(その4)の説明図



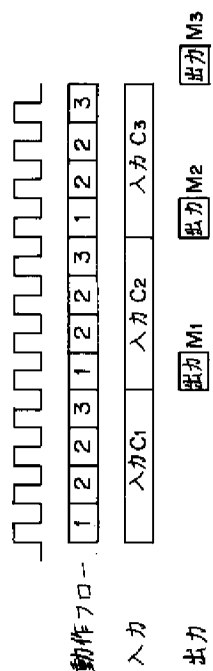
[Drawing 10]

## 第1の実施形態におけるCFBモード復号動作

(その1)の説明図

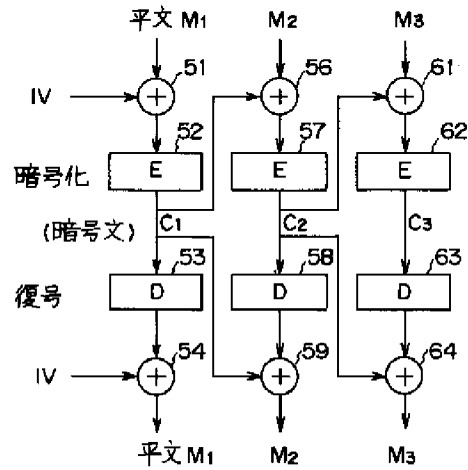


[Drawing 21]

CBCモード復号動作の  
タイミングチャート

[Drawing 24]

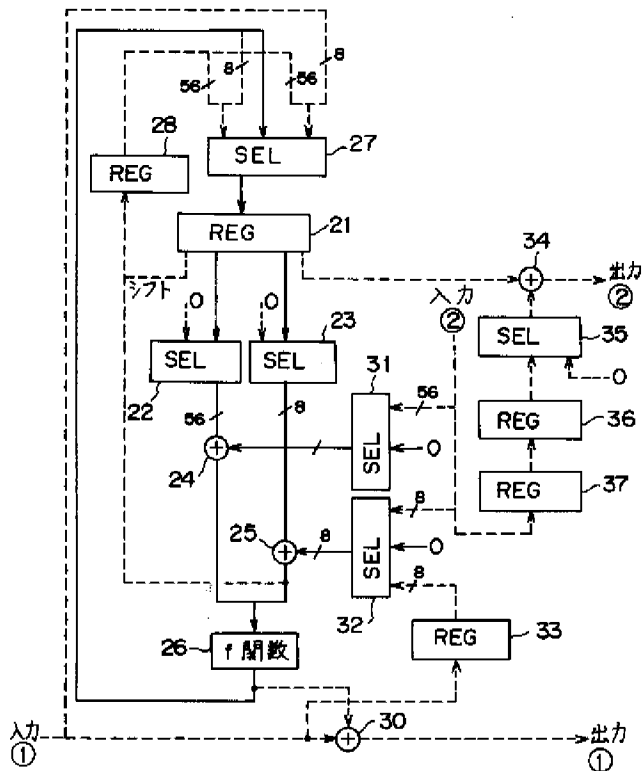
CBCモードにおける暗号化処理の基本説明図



[Drawing 11]

第1の実施形態におけるCFBモード復号動作

(その2)の説明図

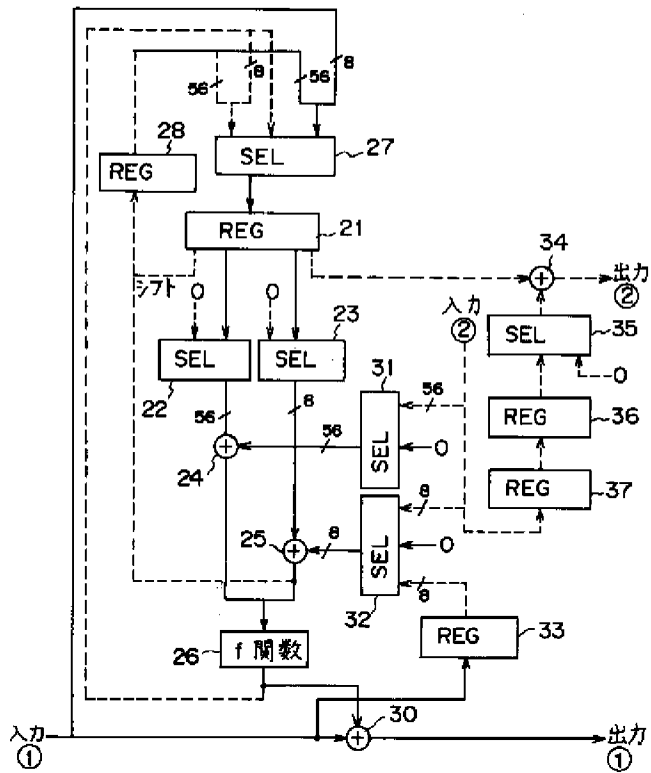


[Drawing 12]



## 第1の実施形態におけるCFBモード復号動作

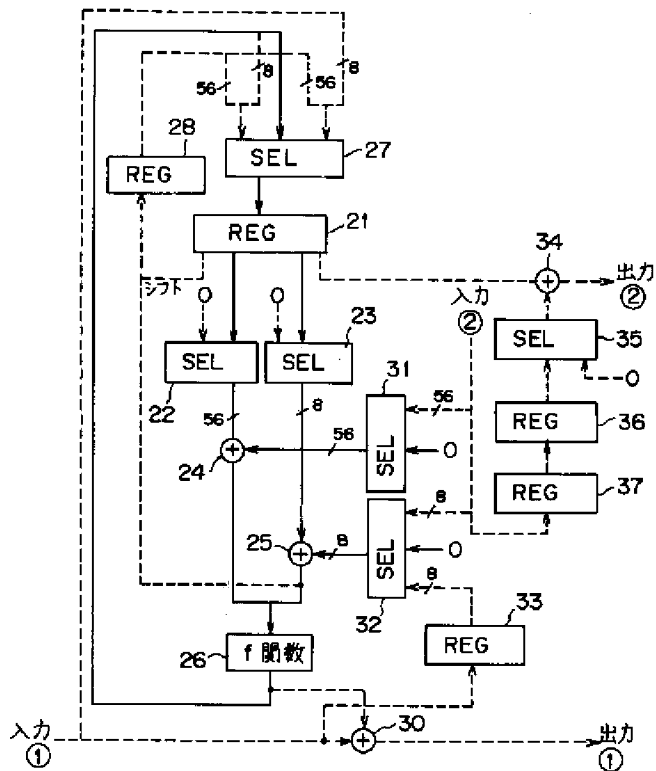
(その3)の説明図



[Drawing 14]

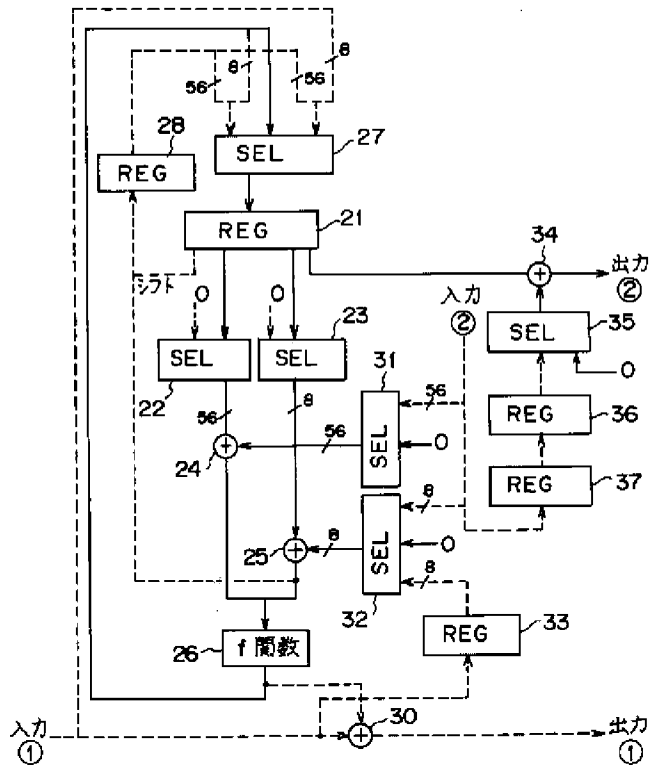


第1の実施形態におけるCBCモード暗号化動作  
(その2)の説明図



[Drawing 16]

第1の実施形態におけるCBCモード暗号化動作  
(その3)の説明図

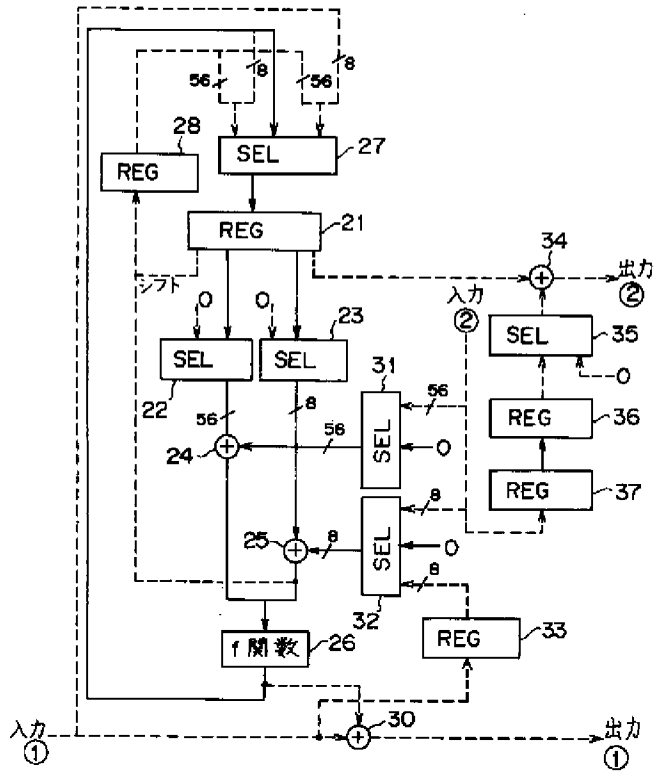


[Drawing 18]



### 第1の実施形態におけるCBCモード復号動作

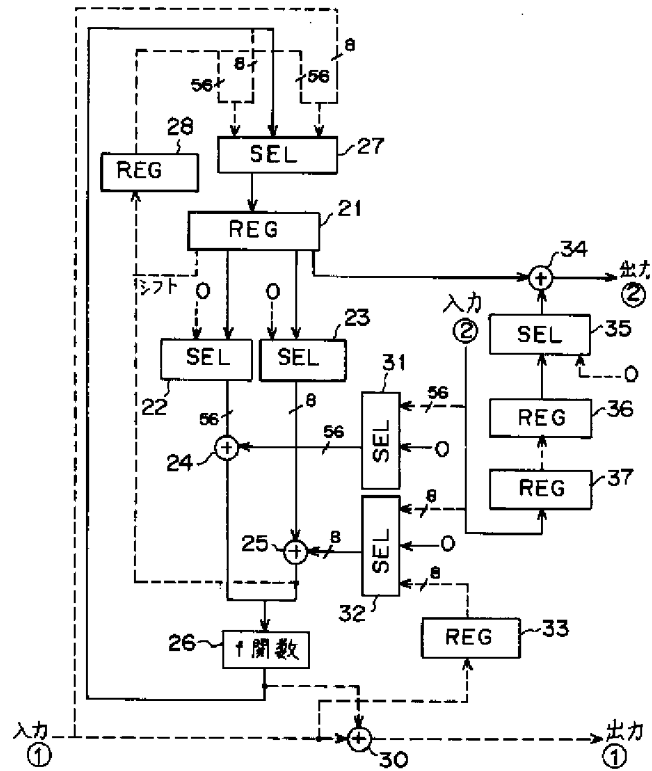
(その2)の説明図



[Drawing 20]

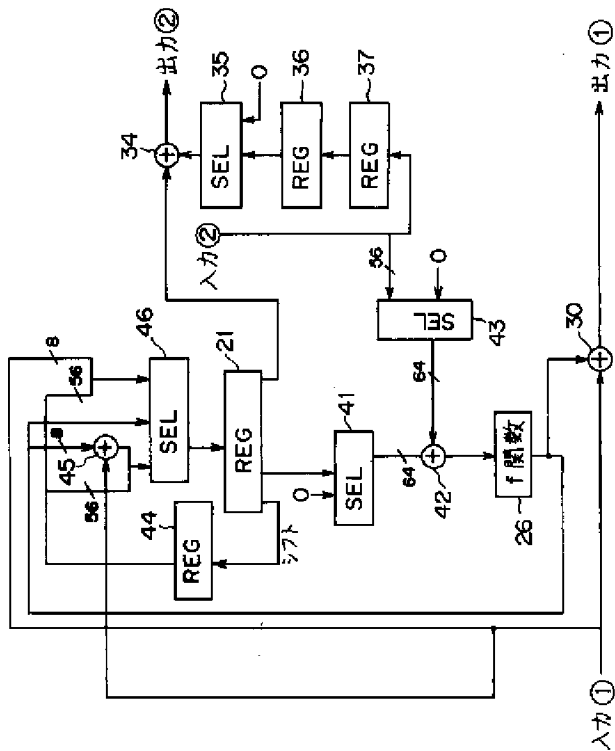
## 第1の実施形態におけるCBCモード復号動作

(その3)の説明図



[Drawing 22]

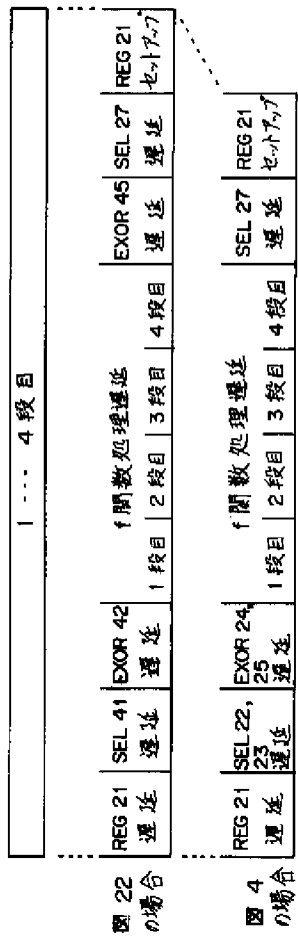
本発明の第2の実施形態としての  
暗号処理回路の構成を示すブロック図



[Drawing 23]

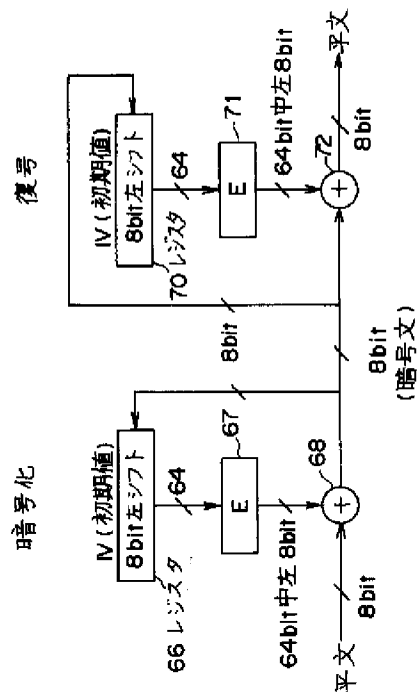


第1の実施形態と第2の実施形態における  
1つのクロック内で行われる処理の比較を示す図



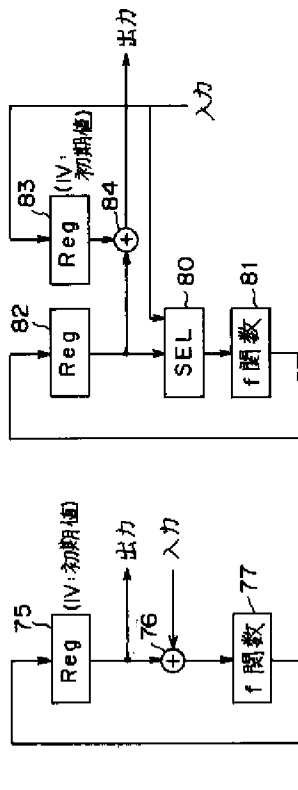
[Drawing 25]

## CFBモードの暗号化処理の基本説明図



[Drawing 26]

CBCモードの暗号処理回路の従来例を示す図

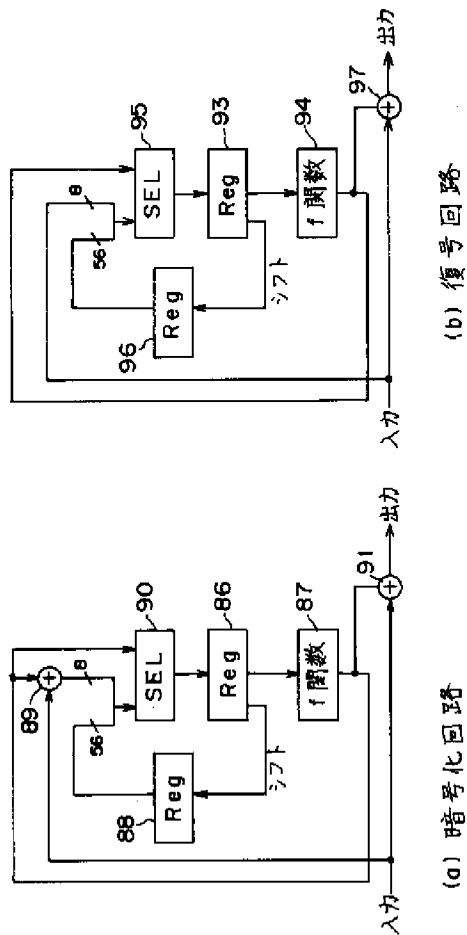


(b) 復号回路

(a) 暗号化回路

[Drawing 27]

CFBモードの暗号処理回路の従来例を示す図



[Translation done.]